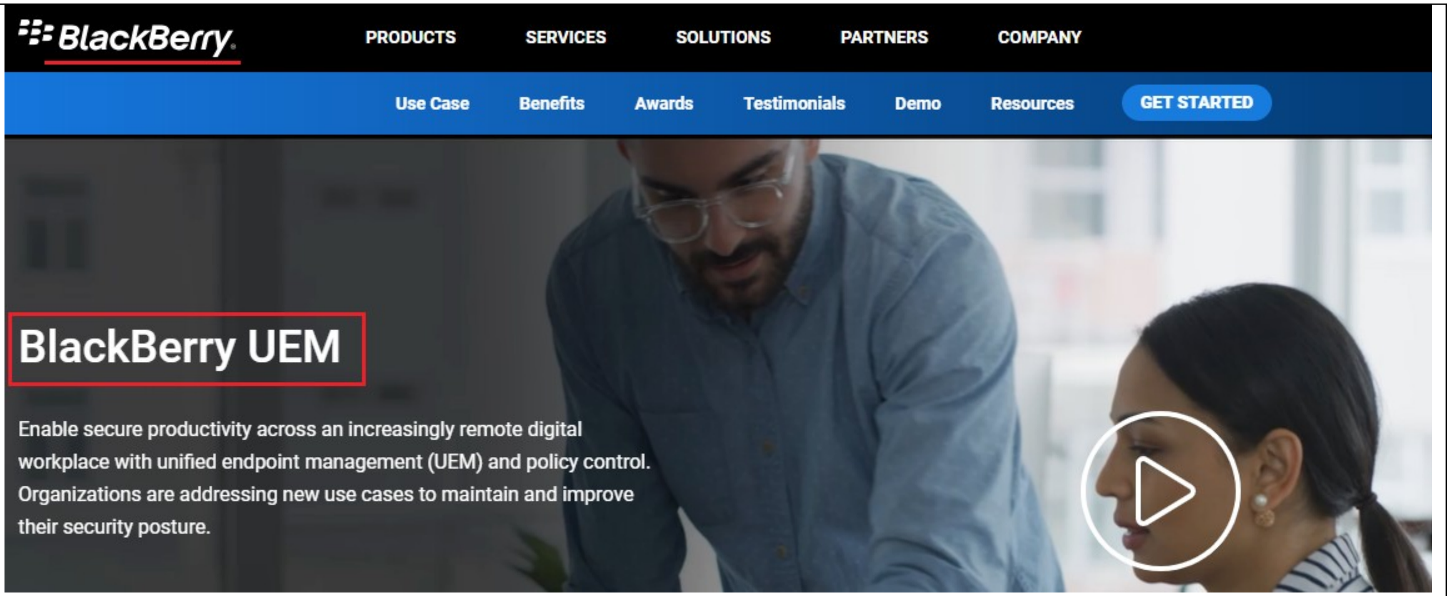
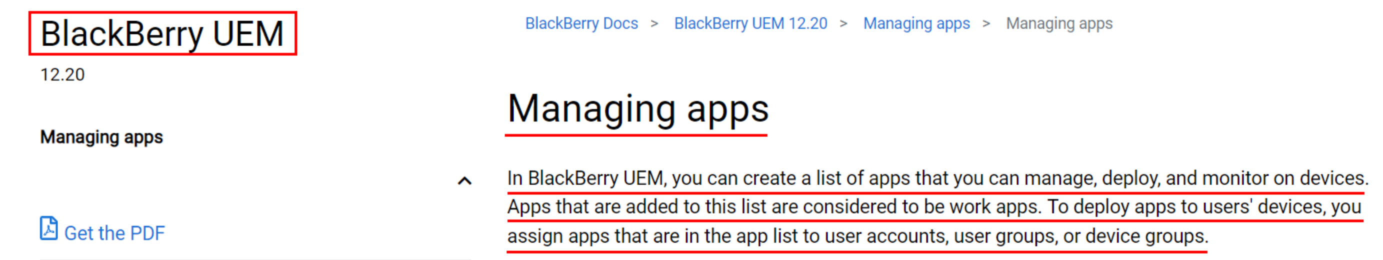
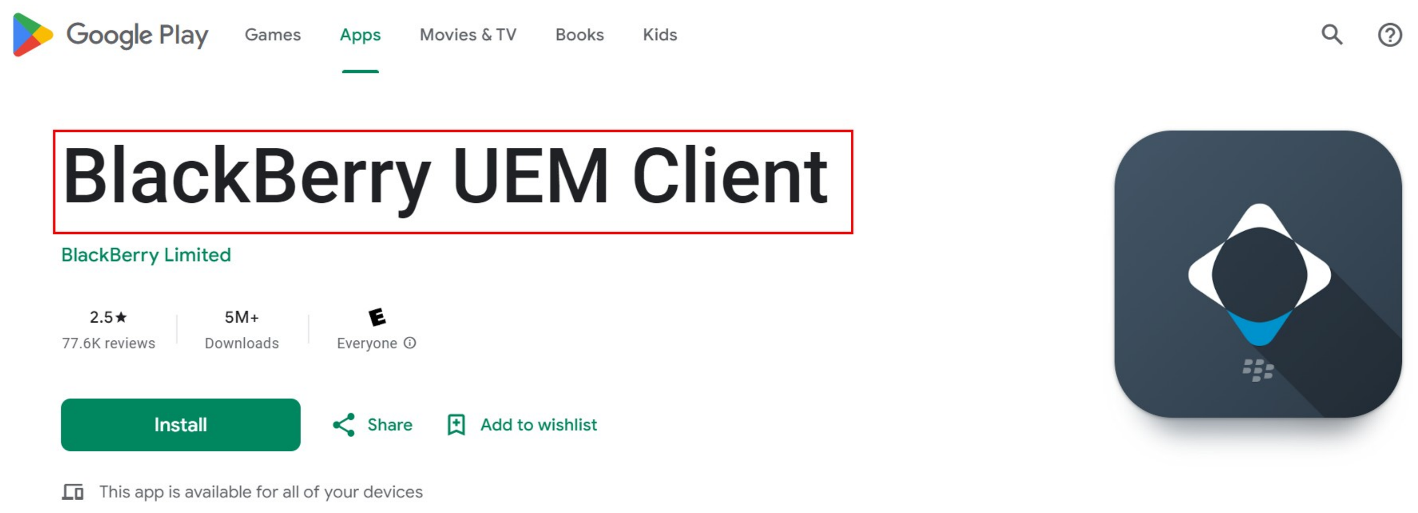
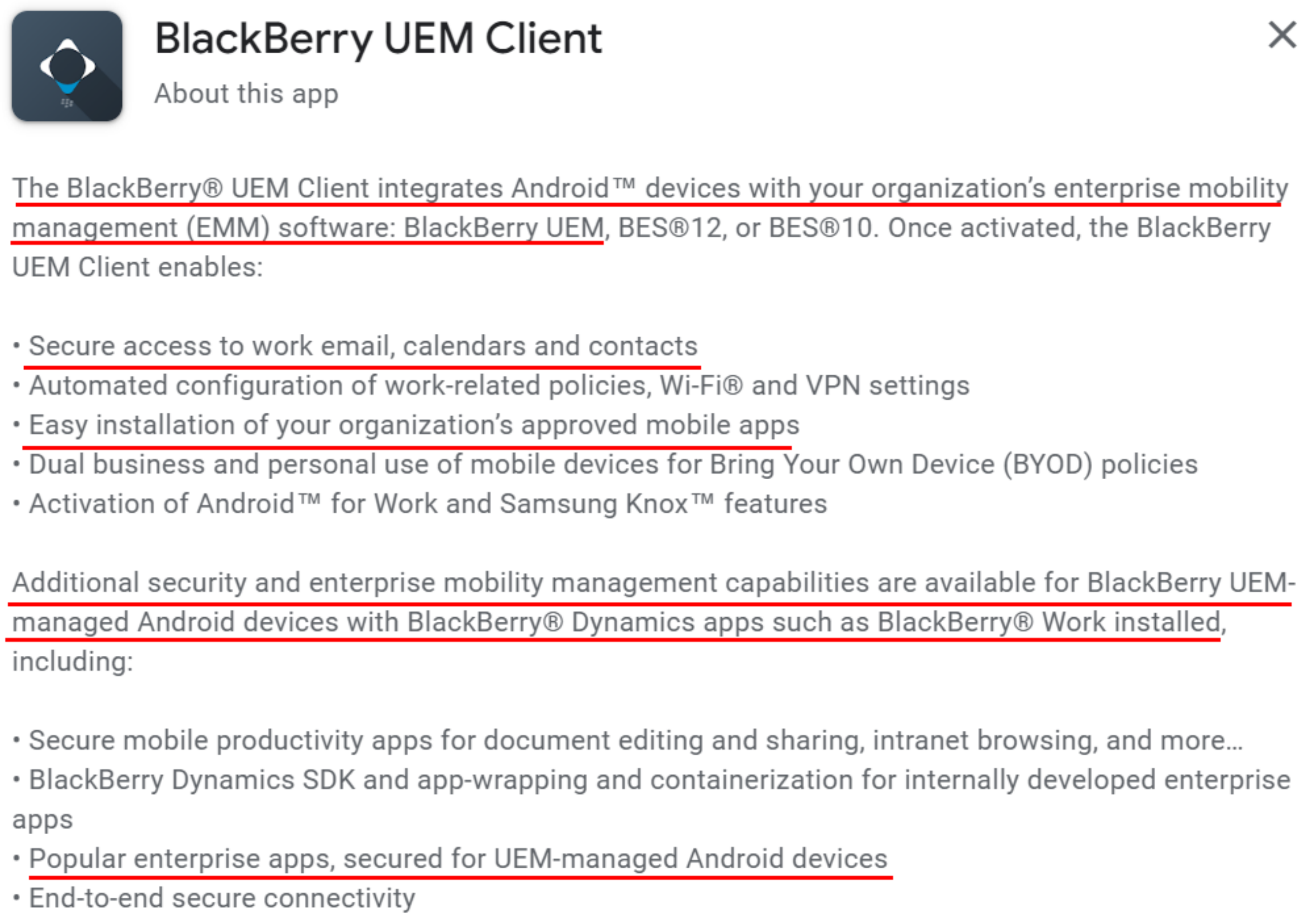
Exhibit 2

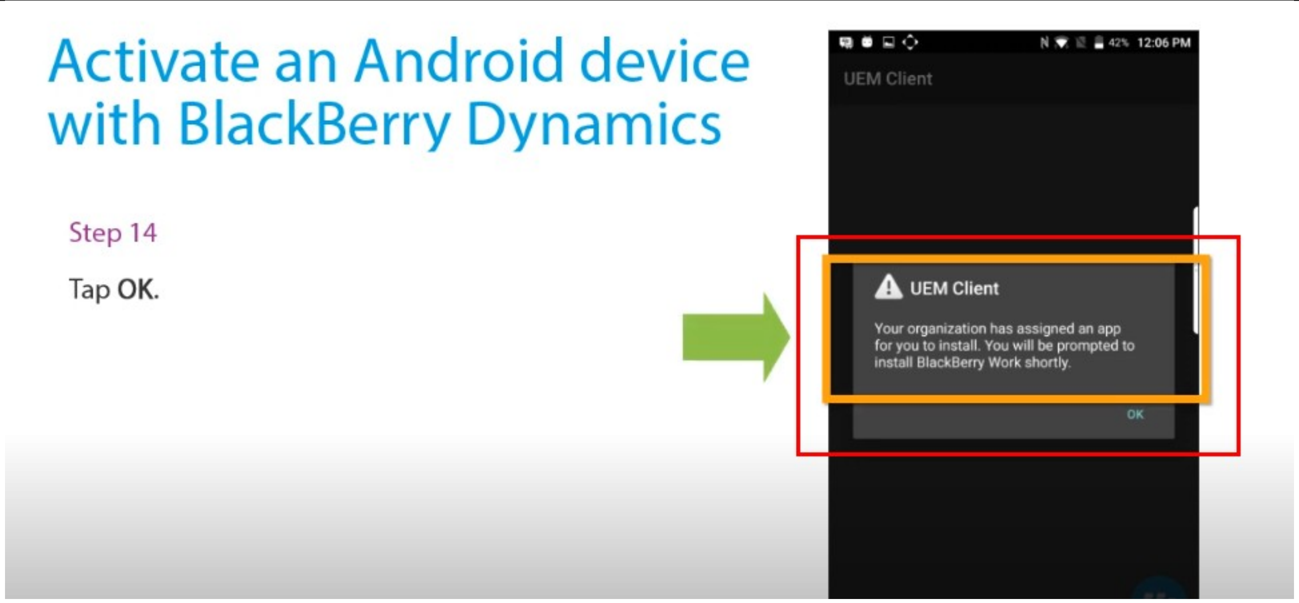
Charted claim:
Method claim:29

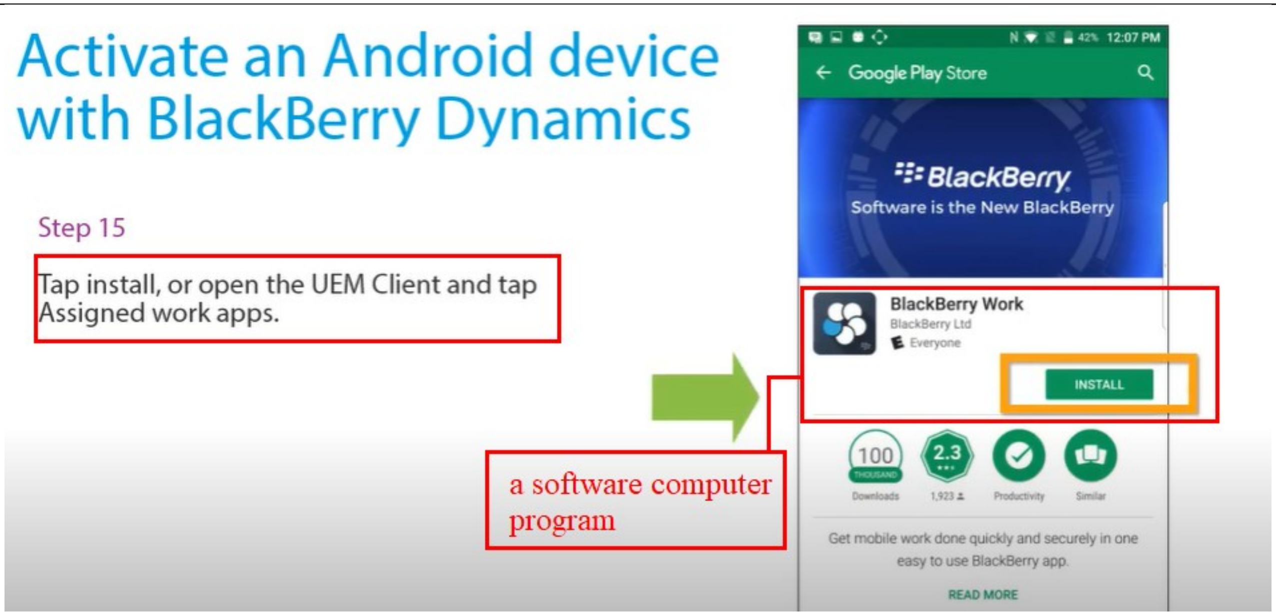
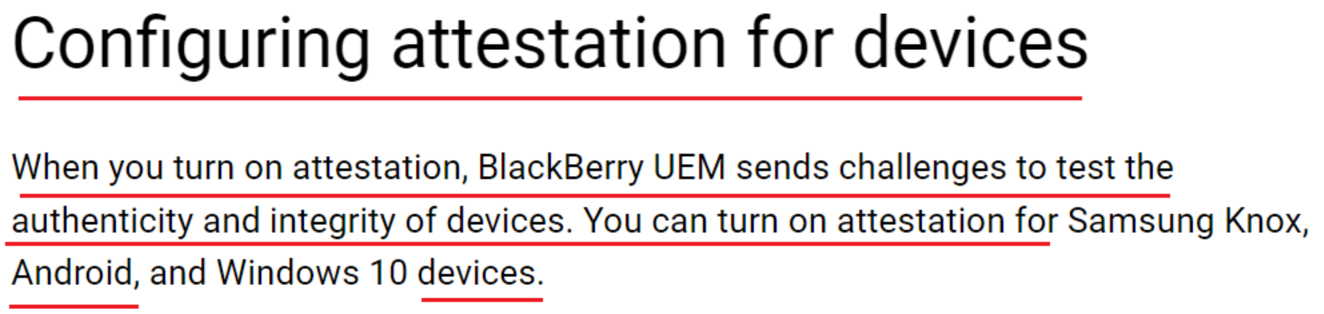
US7895643B2	Blackberry UEM ("The accused instrumentality")
29. A method for assuring proper execution of a software computer program executing on a computing system, the method comprising:	The accused instrumentality discloses a method for assuring proper execution of a software computer program (e.g., a work application, etc.) executing on a computing system (e.g., a user's device running the application, etc.). As shown below, the accused instrumentality is a Unified Endpoint Management (UEM) solution that allows enterprises to manage and control devices linked to employees. It provides secure operating, installing and executing of work applications for the linked devices. Further, the accused instrumentality utilizes device attestation for verifying device & app integrity. When execution of a work application (e.g., a software computer program) running on an employee device requires integrity verification, it utilizes device attestation feature for the verification.

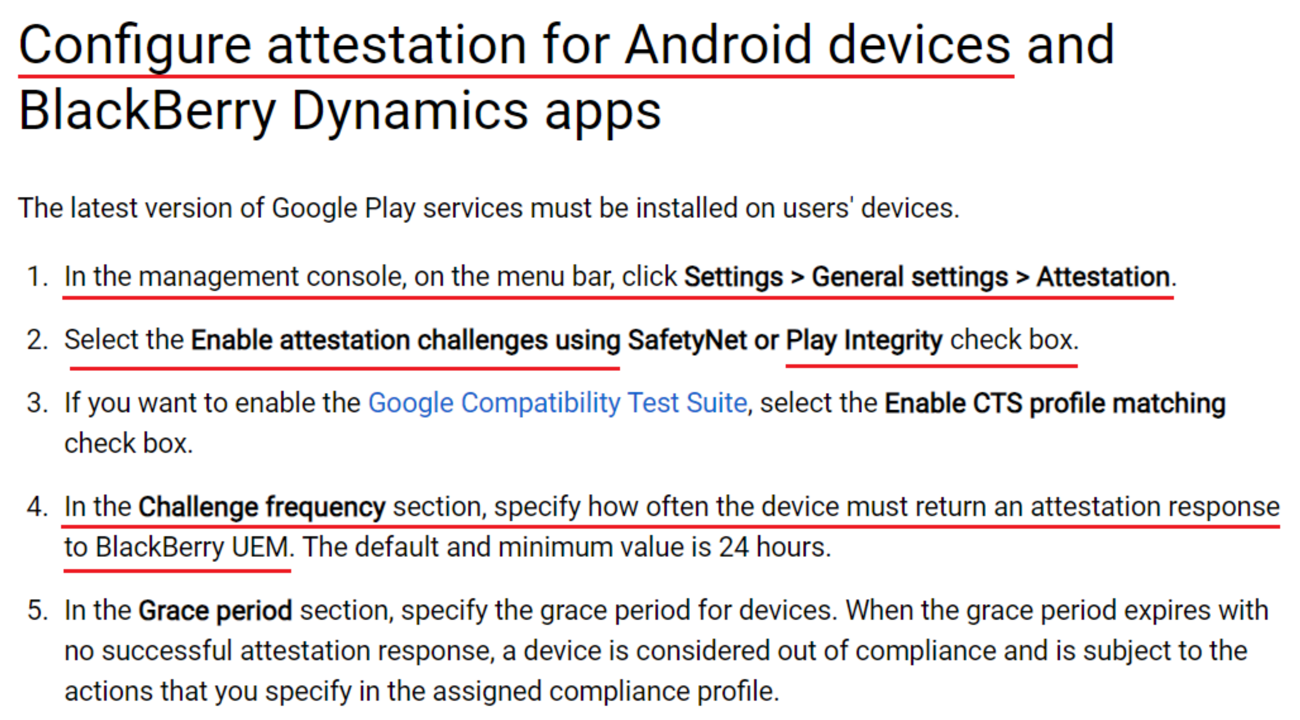
	 BlackBerry PRODUCTS SERVICES SOLUTIONS PARTNERS COMPANY Use Case Benefits Awards Testimonials Demo Resources GET STARTED BlackBerry UEM Enable secure productivity across an increasingly remote digital workplace with unified endpoint management (UEM) and policy control. Organizations are addressing new use cases to maintain and improve their security posture. https://www.blackberry.com/us/en/products/blackberry-uem  BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps Managing apps In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

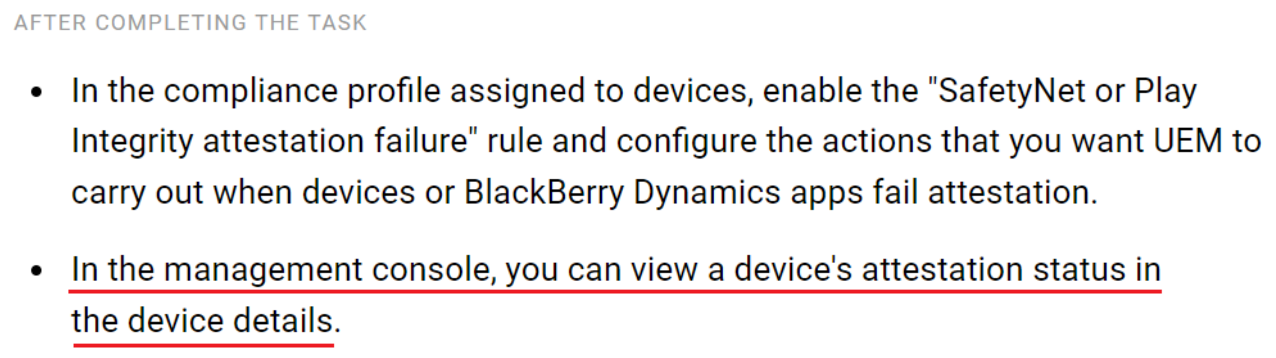
	You use the BlackBerry UEM Client to activate your device for work. When you activate your device, the device is associated with BlackBerry UEM and is granted access to work data and the productivity apps that your administrator assigned to your device. Your administrator determines the degree of protection for your device based on your role and assigns IT policies and profiles to make sure the appropriate device features are available to you and to secure work data on your device. https://docs.blackberry.com/en/endpoint-management/blackberry-uem-client/latest#:~:text=You%20use%20the%20BlackBerry%20UEM,work%20data%20on%20your%20device.  BlackBerry UEM Client BlackBerry Limited 2.5★ 77.6K reviews · 5M+ Downloads · E Everyone This app is available for all of your devices https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

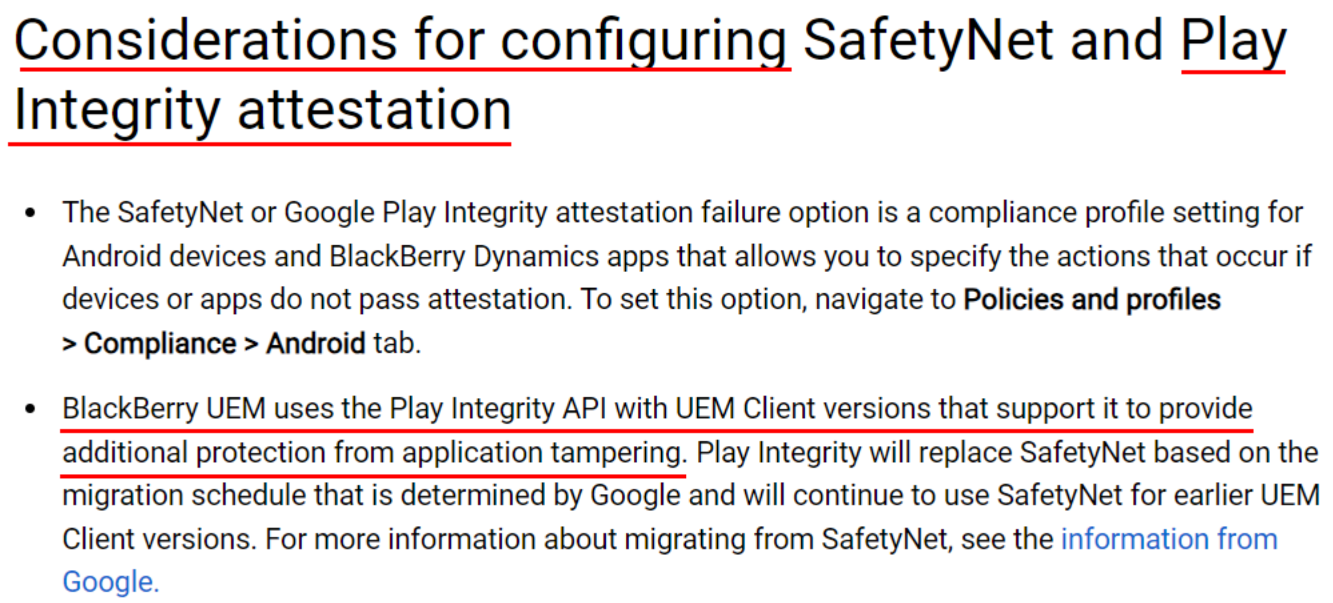
	 BlackBerry UEM Client About this app The BlackBerry® UEM Client integrates Android™ devices with your organization's enterprise mobility management (EMM) software: BlackBerry UEM, BES®12, or BES®10. Once activated, the BlackBerry UEM Client enables: • Secure access to work email, calendars and contacts • Automated configuration of work-related policies, Wi-Fi® and VPN settings • Easy installation of your organization's approved mobile apps • Dual business and personal use of mobile devices for Bring Your Own Device (BYOD) policies • Activation of Android™ for Work and Samsung Knox™ features Additional security and enterprise mobility management capabilities are available for BlackBerry UEM-managed Android devices with BlackBerry® Dynamics apps such as BlackBerry® Work installed, including: • Secure mobile productivity apps for document editing and sharing, intranet browsing, and more… • BlackBerry Dynamics SDK and app-wrapping and containerization for internally developed enterprise apps • Popular enterprise apps, secured for UEM-managed Android devices • End-to-end secure connectivity https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

	 Activate an Android device with BlackBerry Dynamics Step 14 Tap **OK.** UEM Client Your organization has assigned an app for you to install. You will be prompted to install BlackBerry Work shortly. https://www.youtube.com/watch?v=zRrI1TpLVXM

	 Activate an Android device with BlackBerry Dynamics Step 15 Tap install, or open the UEM Client and tap Assigned work apps. a software computer program Google Play Store BlackBerry Software is the New BlackBerry BlackBerry Work BlackBerry Ltd Everyone INSTALL 100 THOUSAND Downloads 2.3 1,923 Productivity Similar READ MORE Get mobile work done quickly and securely in one easy to use BlackBerry app. https://www.youtube.com/watch?v=zRrI1TpLVXM The accused instrumentality utilizes an attestation API such as Play Integrity API for an android device of an employee. The work app sends a request to the attestation API, it sends a signed token back to the app. The app forwards the signed token to a server for verification  Configuring attestation for devices When you turn on attestation, BlackBerry UEM sends challenges to test the authenticity and integrity of devices. You can turn on attestation for Samsung Knox, Android, and Windows 10 devices.

	https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation  **Configure attestation for Android devices and BlackBerry Dynamics apps** The latest version of Google Play services must be installed on users' devices. 1. In the management console, on the menu bar, click **Settings > General settings > Attestation**. 2. Select the **Enable attestation challenges using SafetyNet or Play Integrity** check box. 3. If you want to enable the Google Compatibility Test Suite, select the **Enable CTS profile matching** check box. 4. In the **Challenge frequency** section, specify how often the device must return an attestation response to BlackBerry UEM. The default and minimum value is 24 hours. 5. In the **Grace period** section, specify the grace period for devices. When the grace period expires with no successful attestation response, a device is considered out of compliance and is subject to the actions that you specify in the assigned compliance profile. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841

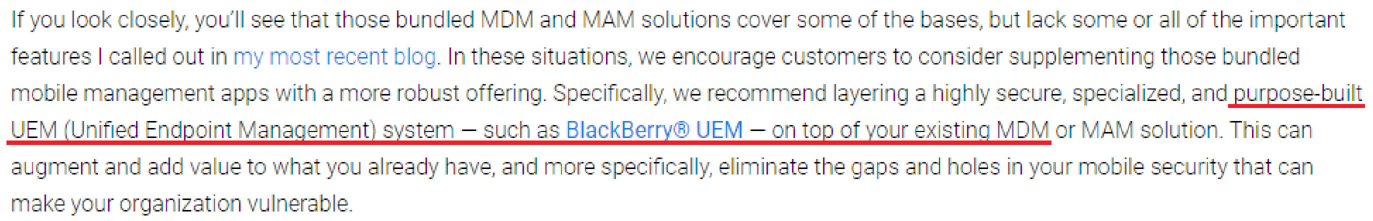
	 AFTER COMPLETING THE TASK • In the compliance profile assigned to devices, enable the "SafetyNet or Play Integrity attestation failure" rule and configure the actions that you want UEM to carry out when devices or BlackBerry Dynamics apps fail attestation. • In the management console, you can view a device's attestation status in the device details. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841



Considerations for configuring SafetyNet and Play Integrity attestation

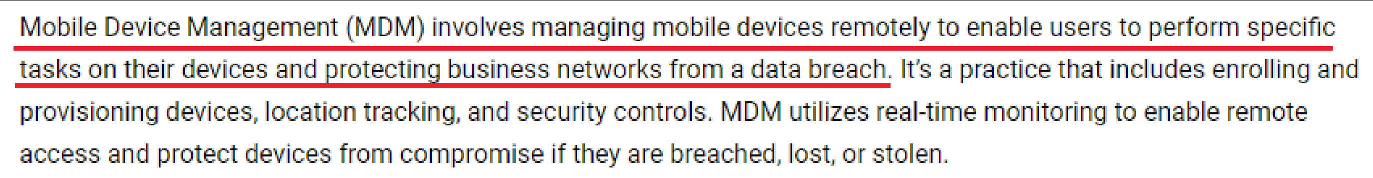
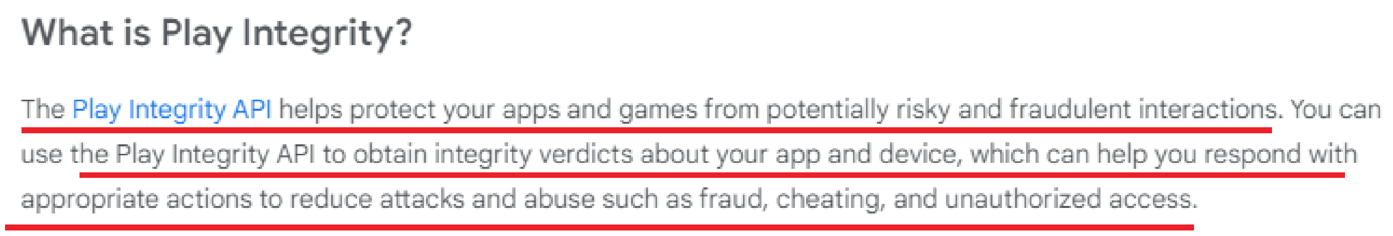
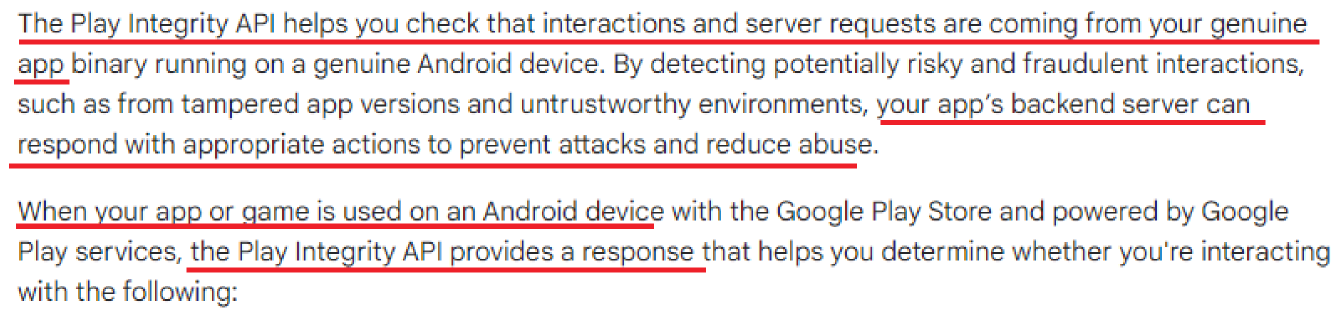
- The SafetyNet or Google Play Integrity attestation failure option is a compliance profile setting for Android devices and BlackBerry Dynamics apps that allows you to specify the actions that occur if devices or apps do not pass attestation. To set this option, navigate to **Policies and profiles > Compliance > Android** tab.
- BlackBerry UEM uses the Play Integrity API with UEM Client versions that support it to provide additional protection from application tampering. Play Integrity will replace SafetyNet based on the migration schedule that is determined by Google and will continue to use SafetyNet for earlier UEM Client versions. For more information about migrating from SafetyNet, see the information from Google.

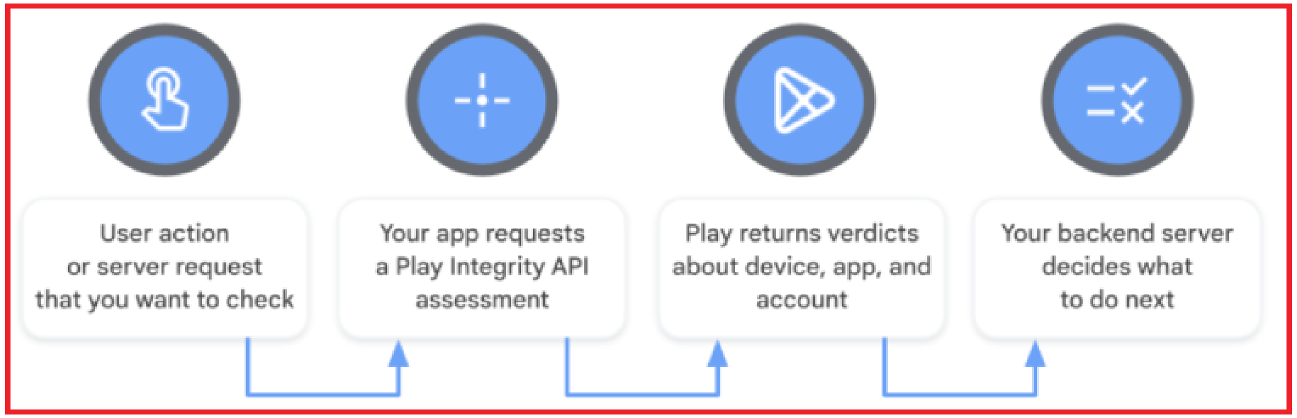
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_18/administration/device-features-it-policies/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/Considerations-for-configuring-attestation-for-Android-devices-and-apps-using-SafetyNet

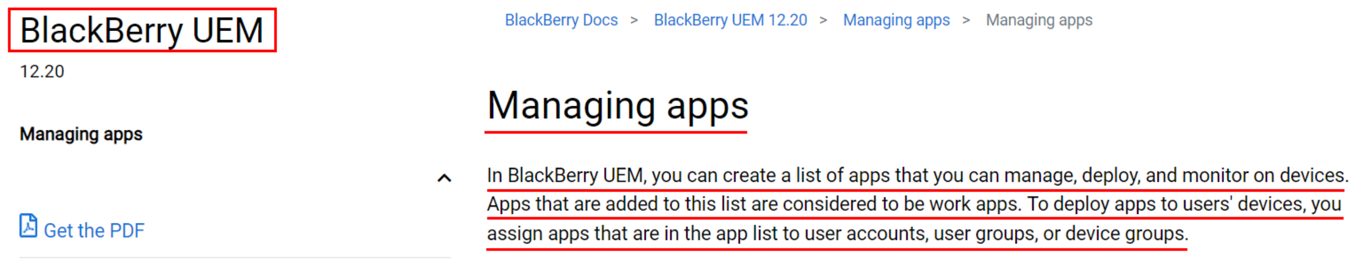


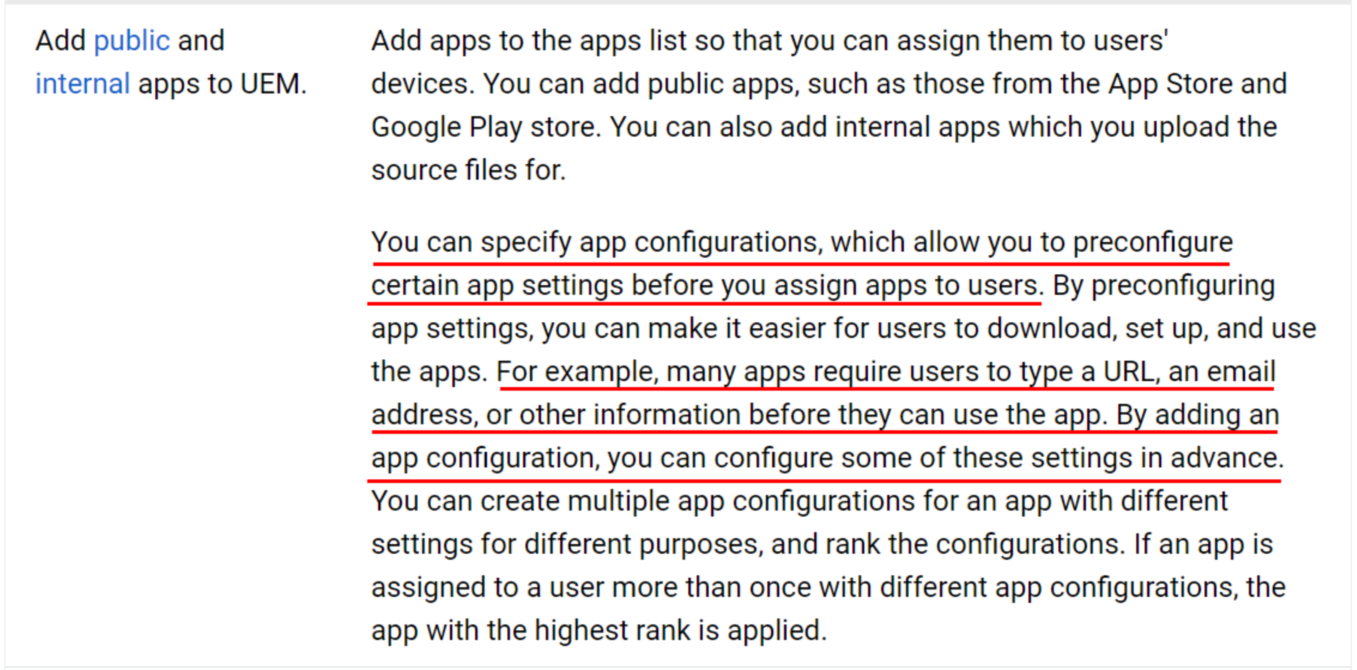
If you look closely, you'll see that those bundled MDM and MAM solutions cover some of the bases, but lack some or all of the important features I called out in my most recent blog. In these situations, we encourage customers to consider supplementing those bundled mobile management apps with a more robust offering. Specifically, we recommend layering a highly secure, specialized, and purpose-built UEM (Unified Endpoint Management) system — such as BlackBerry® UEM — on top of your existing MDM or MAM solution. This can augment and add value to what you already have, and more specifically, eliminate the gaps and holes in your mobile security that can make your organization vulnerable.

https://blogs.blackberry.com/en/2022/10/bundled-mobile-security-is-it-enough

	 Mobile Device Management (MDM) involves managing mobile devices remotely to enable users to perform specific tasks on their devices and protecting business networks from a data breach. It's a practice that includes enrolling and provisioning devices, location tracking, and security controls. MDM utilizes real-time monitoring to enable remote access and protect devices from compromise if they are breached, lost, or stolen. https://www.blackberry.com/us/en/solutions/endpoint-security/unified-endpoint-management/mobile-device-management  What is Play Integrity? The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions. You can use the Play Integrity API to obtain integrity verdicts about your app and device, which can help you respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access. https://developer.android.com/codelabs/add-play-integrity#0  The Play Integrity API helps you check that interactions and server requests are coming from your genuine app binary running on a genuine Android device. By detecting potentially risky and fraudulent interactions, such as from tampered app versions and untrustworthy environments, your app's backend server can respond with appropriate actions to prevent attacks and reduce abuse. When your app or game is used on an Android device with the Google Play Store and powered by Google Play services, the Play Integrity API provides a response that helps you determine whether you're interacting with the following: https://developer.android.com/google/play/integrity/overview

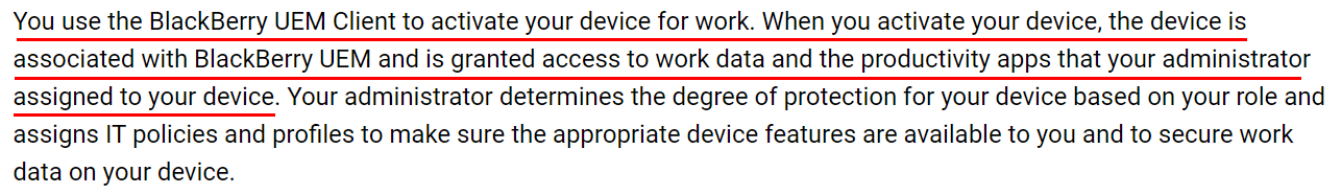
	When a user performs an action in your app, you can call the Play Integrity API to check that it happened in your genuine app binary, installed by Google Play, running on a genuine Android device. You can also opt-in to additional information in the response including the volume of requests a device had made recently and signals about the environment, including the app access risk verdict and the Play Protect verdict. If anything is wrong with the verdicts, then your app's backend server can decide what to do to defend against problems like abuse and fraud, misuse and cheating, unauthorized access, and attacks.  User action or server request that you want to check Your app requests a Play Integrity API assessment Play returns verdicts about device, app, and account Your backend server decides what to do next https://developer.android.com/google/play/integrity/overview 1. The server creates a globally unique value in a way that malicious users cannot predict. For example, a cryptographically-secure random number 128 bits or larger. 2. Your app calls the Play Integrity API, and sets the nonce field to the unique value received by your app server. 3. Your app sends the signed result of the Play Integrity API to your server. 4. Your server verifies that the nonce field in the signed result matches the unique value it previously generated, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1

providing an integrated computer software program comprised of a software application logic module and an operational assurance logic module;	The accused instrumentality discloses providing an integrated computer software program (e.g., a work application along with attestation API, etc.) comprised of a software application logic module (e.g., work application programs, algorithms, etc.) and an operational assurance logic module (e.g., Attestation API, etc.) As shown below, the accused instrumentality is a Unified Endpoint Management (UEM) solution that allows enterprises to manage and control devices linked to employees. It allows administrators to add work applications that are permitted for use by employees on their managed devices. For example, an application such as E-mail (e.g., an integrated computer software program) includes various algorithms and processes (e.g., a software application logic module) to compose and send email to a recipient. Also, the accused instrumentality utilizes device attestation for verifying device & app integrity. When execution of a work application (e.g., a software computer program) running on an employee device requires integrity verification, it utilizes device attestation feature for the verification. It utilizes an attestation API (e.g., an operation assurance logic module) such as Play Integrity API for an android device of an employee. The work app sends a request to the attestation API, it sends a signed token back to the app. The app forwards the signed token to a server for verification.  BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps Managing apps In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps



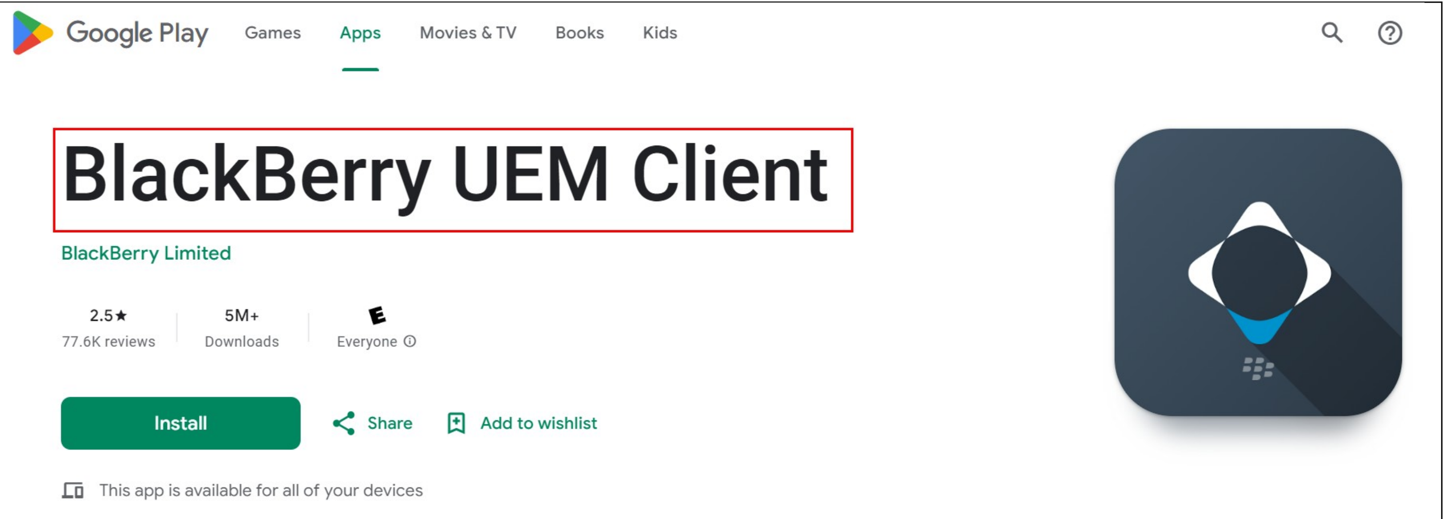
Add public and internal apps to UEM.	Add apps to the apps list so that you can assign them to users' devices. You can add public apps, such as those from the App Store and Google Play store. You can also add internal apps which you upload the source files for. You can specify app configurations, which allow you to preconfigure certain app settings before you assign apps to users. By preconfiguring app settings, you can make it easier for users to download, set up, and use the apps. For example, many apps require users to type a URL, an email address, or other information before they can use the app. By adding an app configuration, you can configure some of these settings in advance. You can create multiple app configurations for an app with different settings for different purposes, and rank the configurations. If an app is assigned to a user more than once with different app configurations, the app with the highest rank is applied.

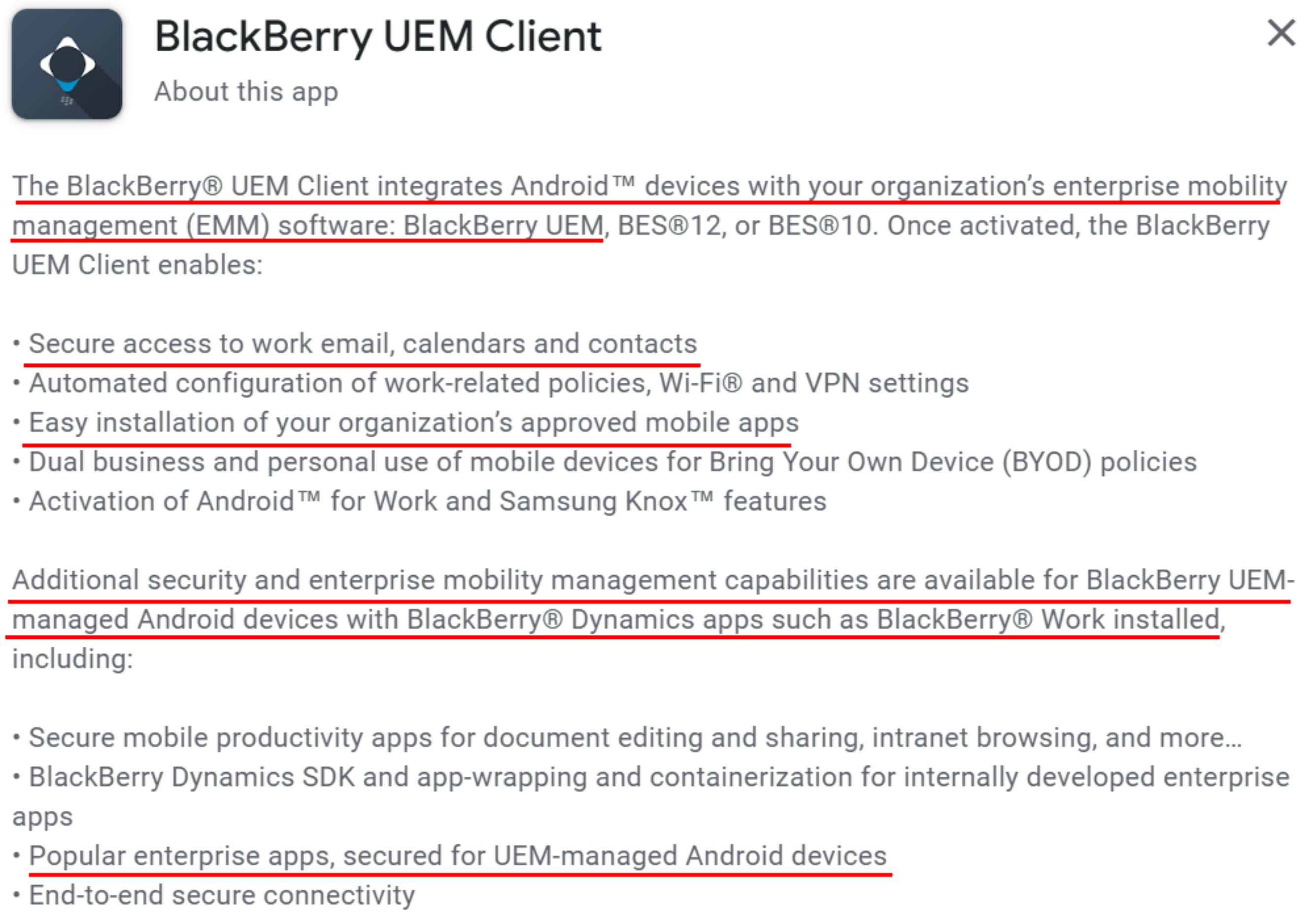
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

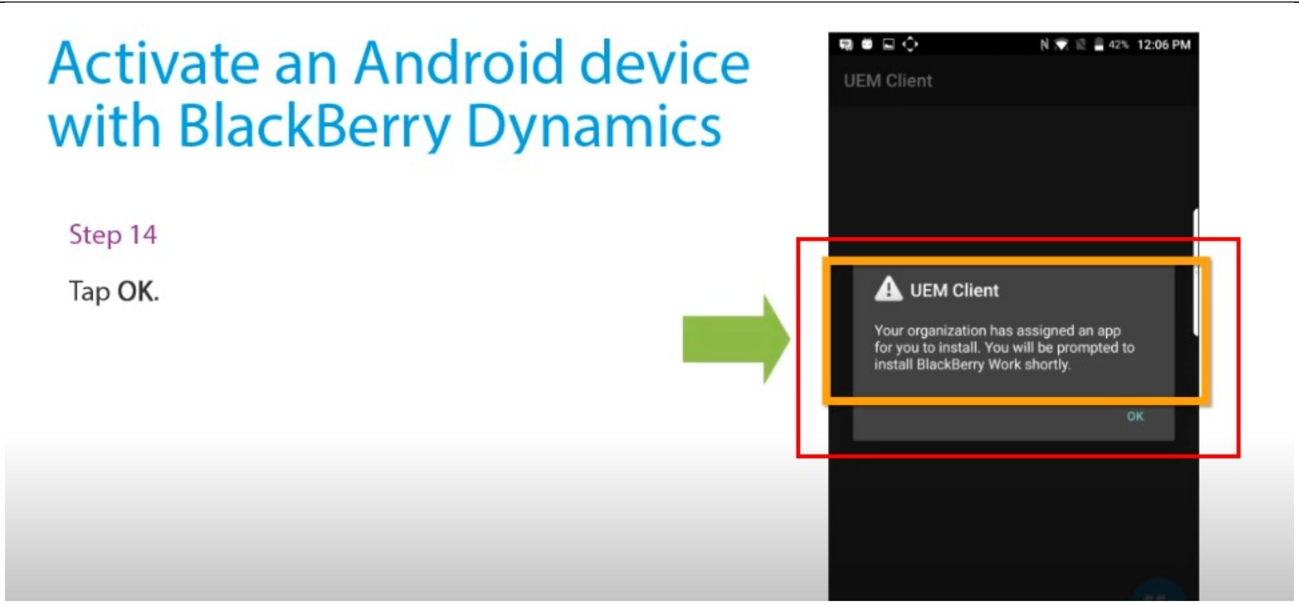


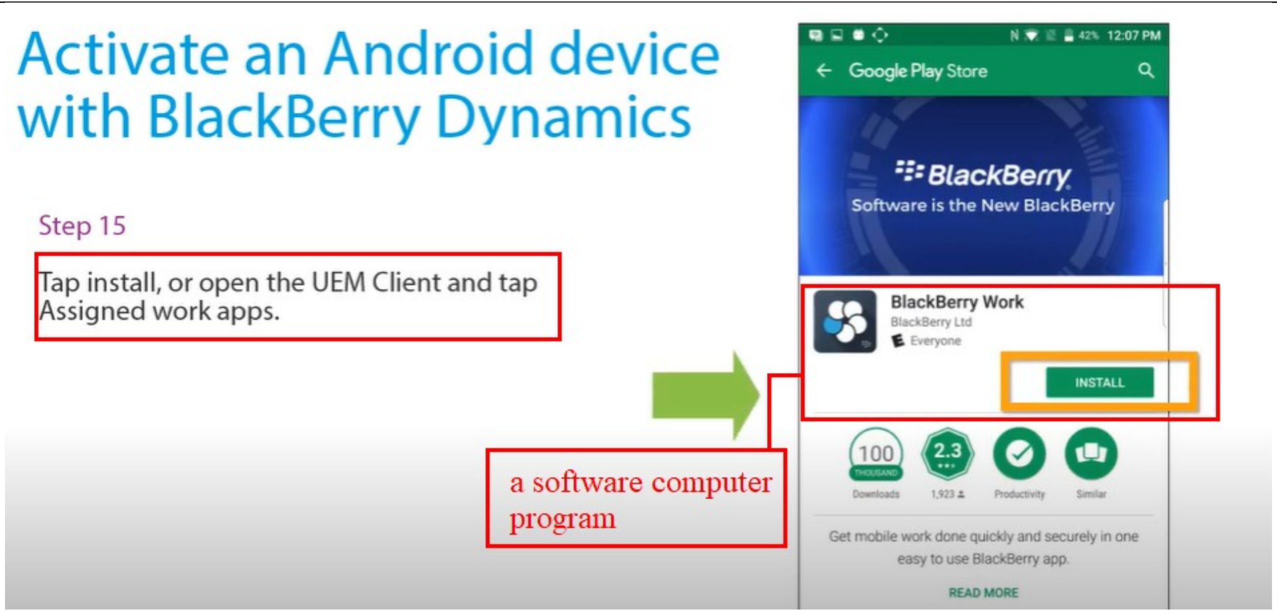
You use the BlackBerry UEM Client to activate your device for work. When you activate your device, the device is associated with BlackBerry UEM and is granted access to work data and the productivity apps that your administrator assigned to your device. Your administrator determines the degree of protection for your device based on your role and assigns IT policies and profiles to make sure the appropriate device features are available to you and to secure work data on your device.

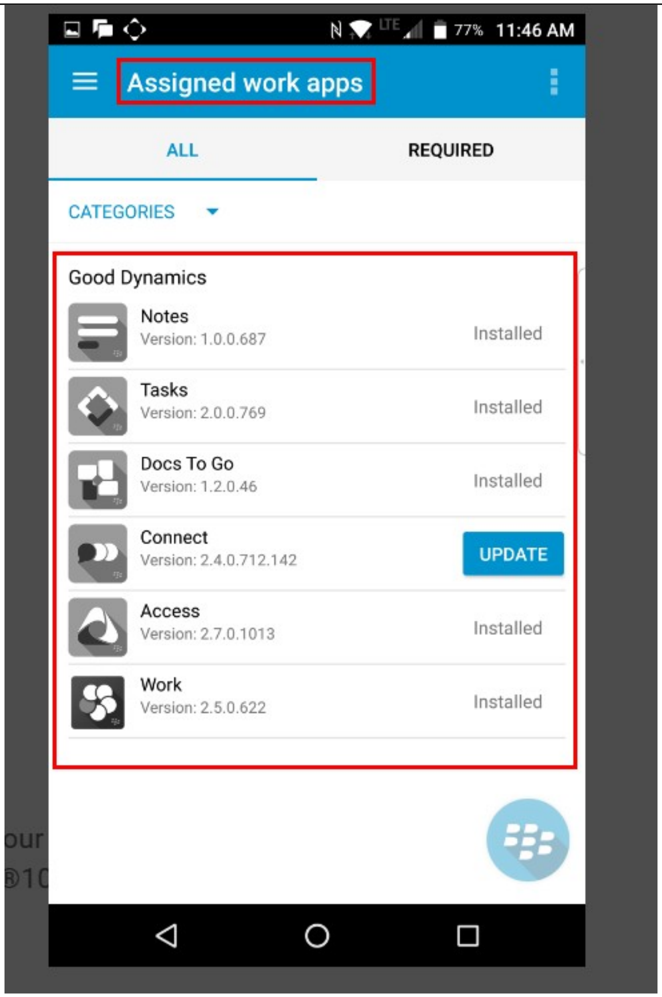
https://docs.blackberry.com/en/endpoint-management/blackberry-uem-client/latest#:~:text=You%20use%20the%20BlackBerry%20UEM,work%20data%20on%20your%20device.

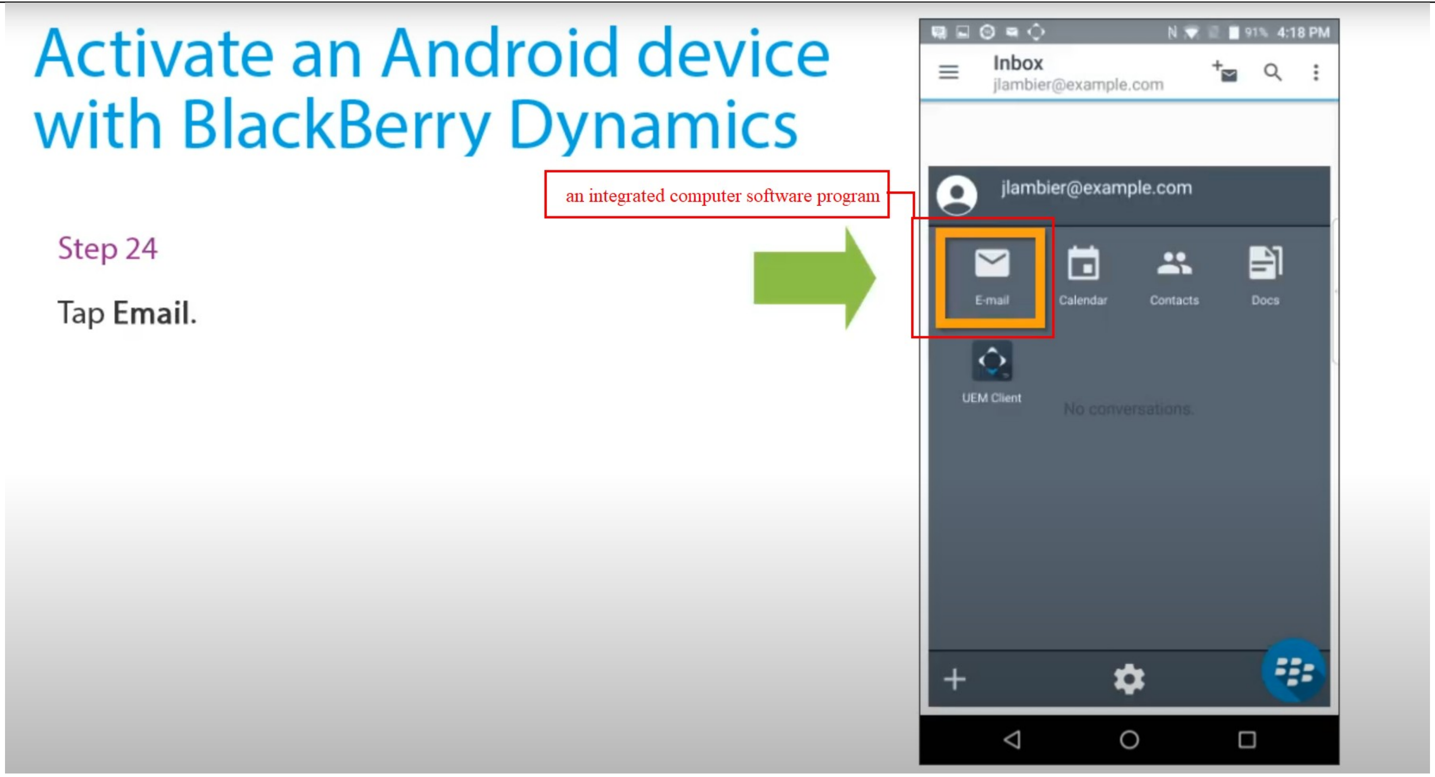
	 Google Play · Games · Apps · Movies & TV · Books · Kids BlackBerry UEM Client BlackBerry Limited 2.5★ 77.6K reviews · 5M+ Downloads · Everyone Install · Share · Add to wishlist This app is available for all of your devices https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

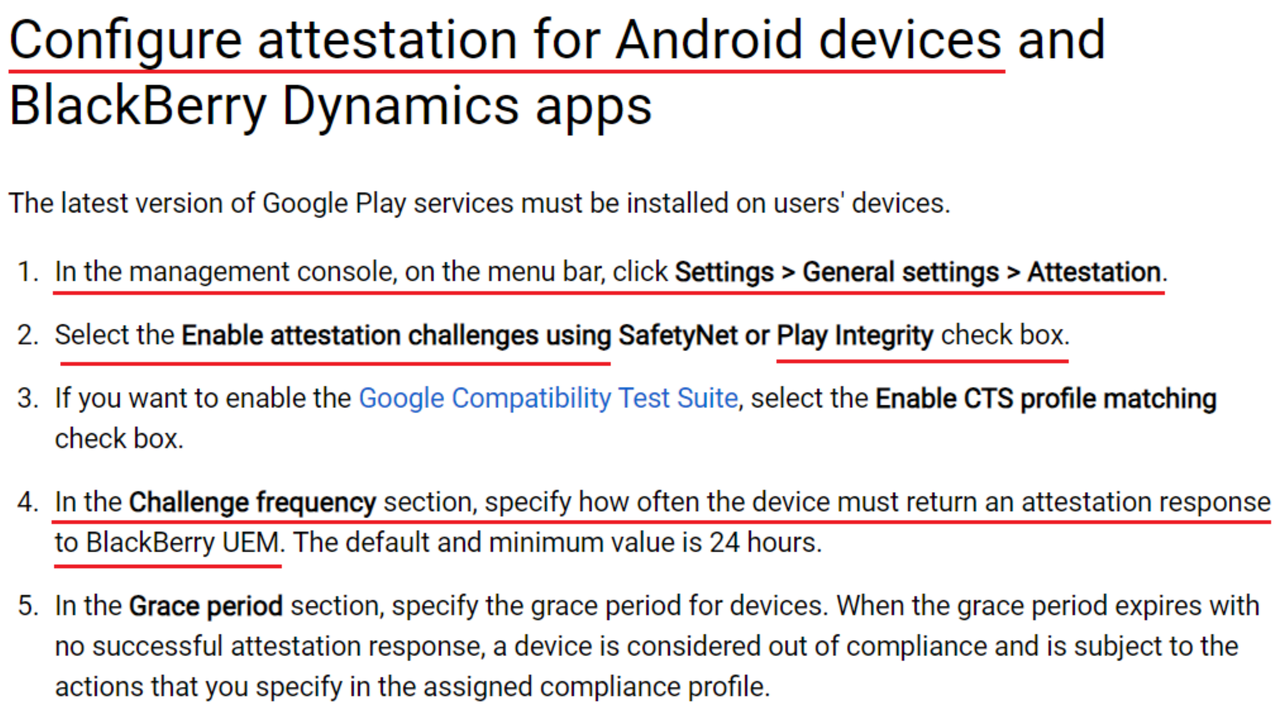
	 BlackBerry UEM Client About this app The BlackBerry® UEM Client integrates Android™ devices with your organization's enterprise mobility management (EMM) software: BlackBerry UEM, BES®12, or BES®10. Once activated, the BlackBerry UEM Client enables: • Secure access to work email, calendars and contacts • Automated configuration of work-related policies, Wi-Fi® and VPN settings • Easy installation of your organization's approved mobile apps • Dual business and personal use of mobile devices for Bring Your Own Device (BYOD) policies • Activation of Android™ for Work and Samsung Knox™ features Additional security and enterprise mobility management capabilities are available for BlackBerry UEM-managed Android devices with BlackBerry® Dynamics apps such as BlackBerry® Work installed, including: • Secure mobile productivity apps for document editing and sharing, intranet browsing, and more… • BlackBerry Dynamics SDK and app-wrapping and containerization for internally developed enterprise apps • Popular enterprise apps, secured for UEM-managed Android devices • End-to-end secure connectivity https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

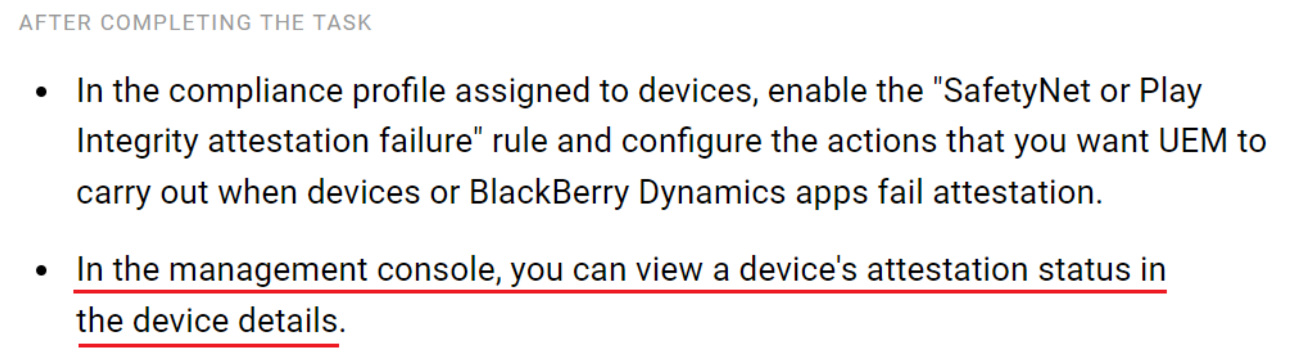
	 Activate an Android device with BlackBerry Dynamics Step 14 Tap **OK**. UEM Client UEM Client Your organization has assigned an app for you to install. You will be prompted to install BlackBerry Work shortly. OK https://www.youtube.com/watch?v=zRrI1TpLVXM

	 Activate an Android device with BlackBerry Dynamics Step 15 Tap install, or open the UEM Client and tap Assigned work apps. a software computer program Google Play Store BlackBerry Software is the New BlackBerry BlackBerry Work BlackBerry Ltd Everyone INSTALL 100 THOUSAND Downloads 2.3 1,923 Productivity Similar READ MORE Get mobile work done quickly and securely in one easy to use BlackBerry app. https://www.youtube.com/watch?v=zRrI1TpLVXM

	 Assigned work apps ALL REQUIRED CATEGORIES Good Dynamics Notes Version: 1.0.0.687 Installed Tasks Version: 2.0.0.769 Installed Docs To Go Version: 1.2.0.46 Installed Connect Version: 2.4.0.712.142 UPDATE Access Version: 2.7.0.1013 Installed Work Version: 2.5.0.622 Installed https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

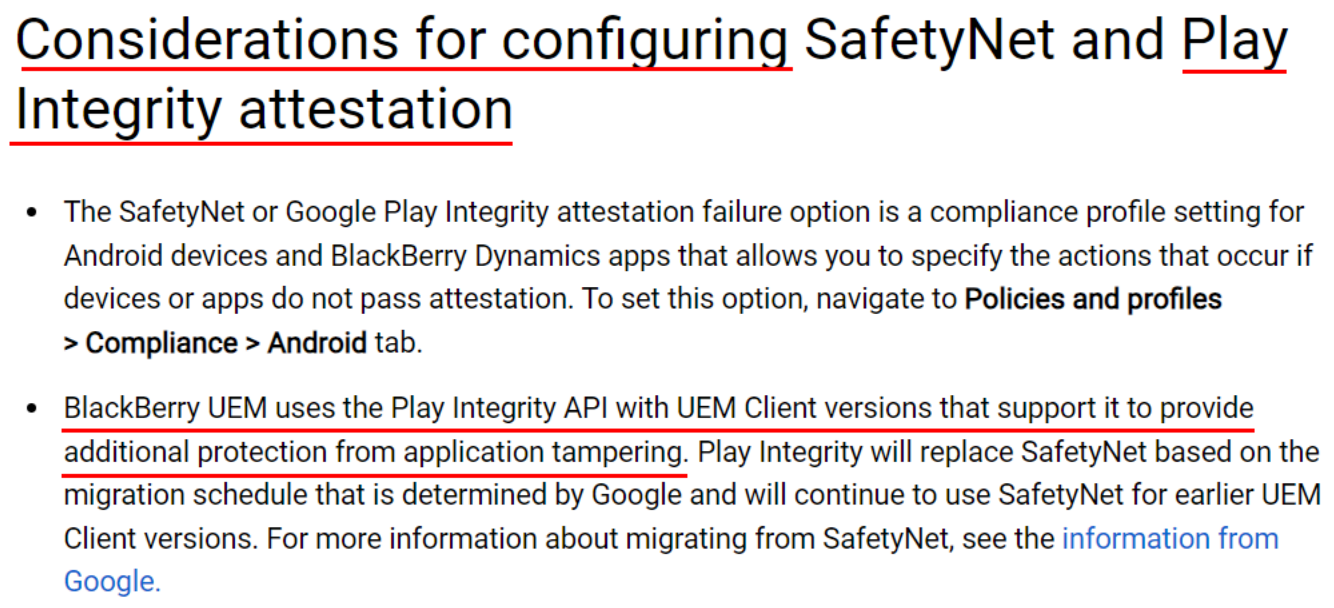
	 Activate an Android device with BlackBerry Dynamics Step 24 Tap **Email.** an integrated computer software program Inbox jlambier@example.com jlambier@example.com E-mail Calendar Contacts Docs UEM Client No conversations. 91% 4:18 PM https://www.youtube.com/watch?v=zRrI1TpLVXM **Configuring attestation for devices** When you turn on attestation, BlackBerry UEM sends challenges to test the authenticity and integrity of devices. You can turn on attestation for Samsung Knox, Android, and Windows 10 devices.

	https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation  Configure attestation for Android devices and BlackBerry Dynamics apps The latest version of Google Play services must be installed on users' devices. 1. In the management console, on the menu bar, click **Settings > General settings > Attestation**. 2. Select the **Enable attestation challenges using SafetyNet or Play Integrity** check box. 3. If you want to enable the Google Compatibility Test Suite, select the **Enable CTS profile matching** check box. 4. In the **Challenge frequency** section, specify how often the device must return an attestation response to BlackBerry UEM. The default and minimum value is 24 hours. 5. In the **Grace period** section, specify the grace period for devices. When the grace period expires with no successful attestation response, a device is considered out of compliance and is subject to the actions that you specify in the assigned compliance profile. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841

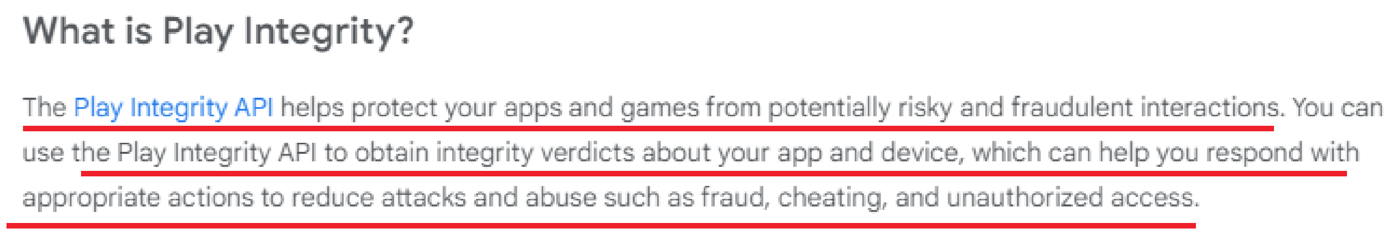
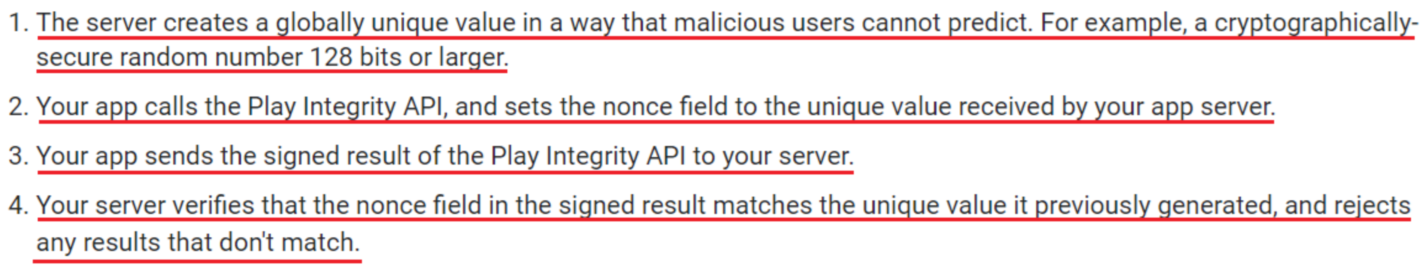

AFTER COMPLETING THE TASK

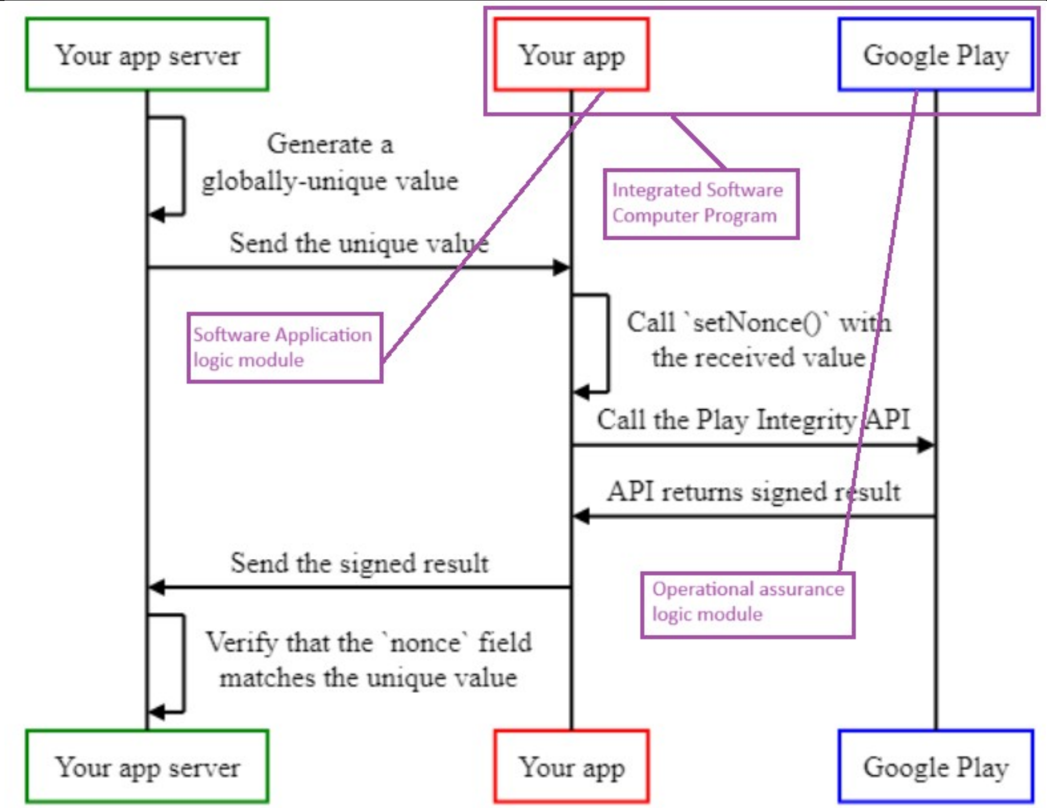
- In the compliance profile assigned to devices, enable the "SafetyNet or Play Integrity attestation failure" rule and configure the actions that you want UEM to carry out when devices or BlackBerry Dynamics apps fail attestation.
- In the management console, you can view a device's attestation status in the device details.

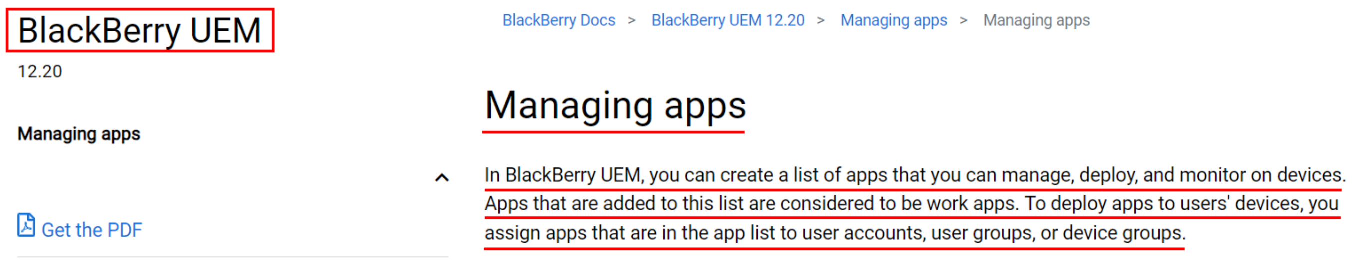
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841

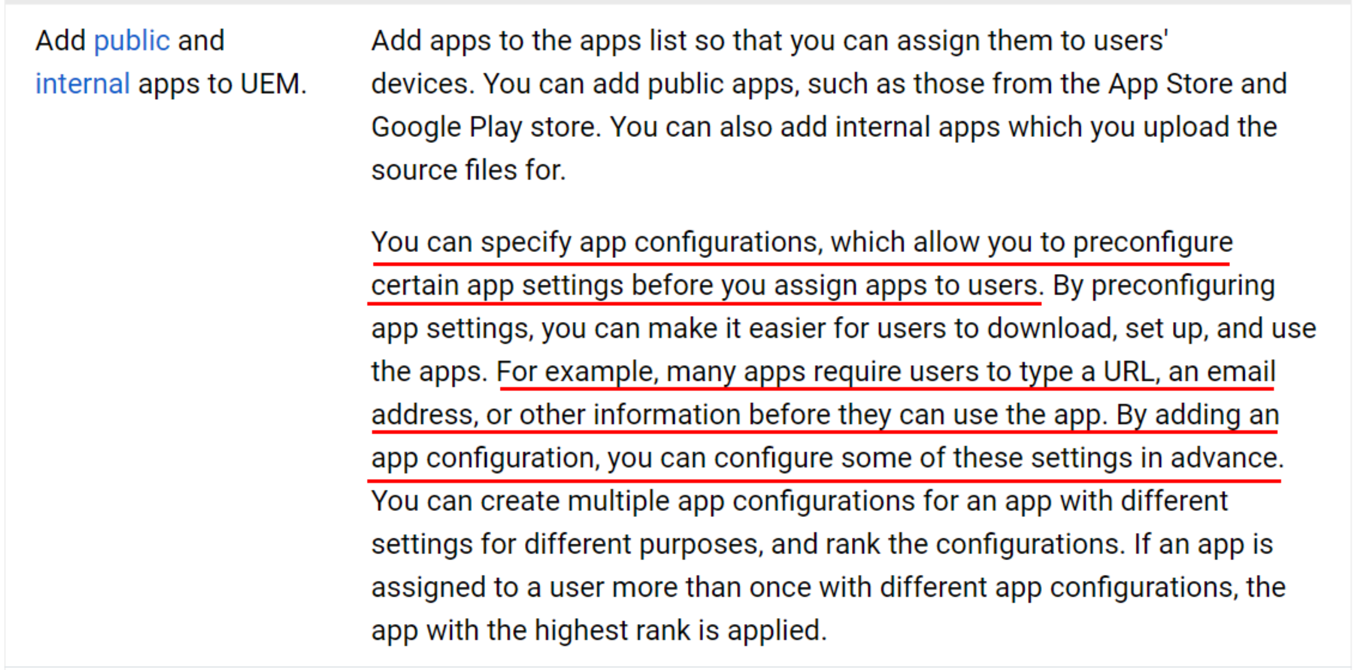
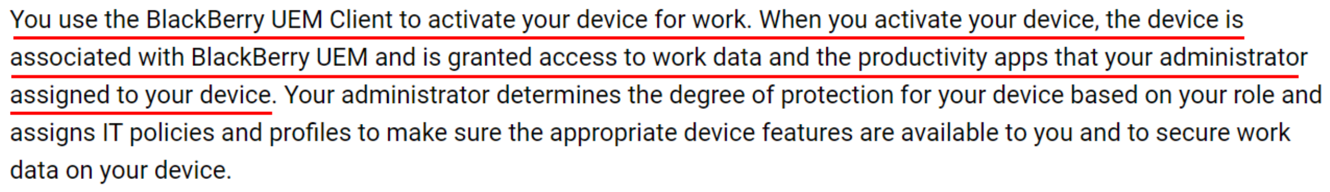

Considerations for configuring SafetyNet and Play Integrity attestation

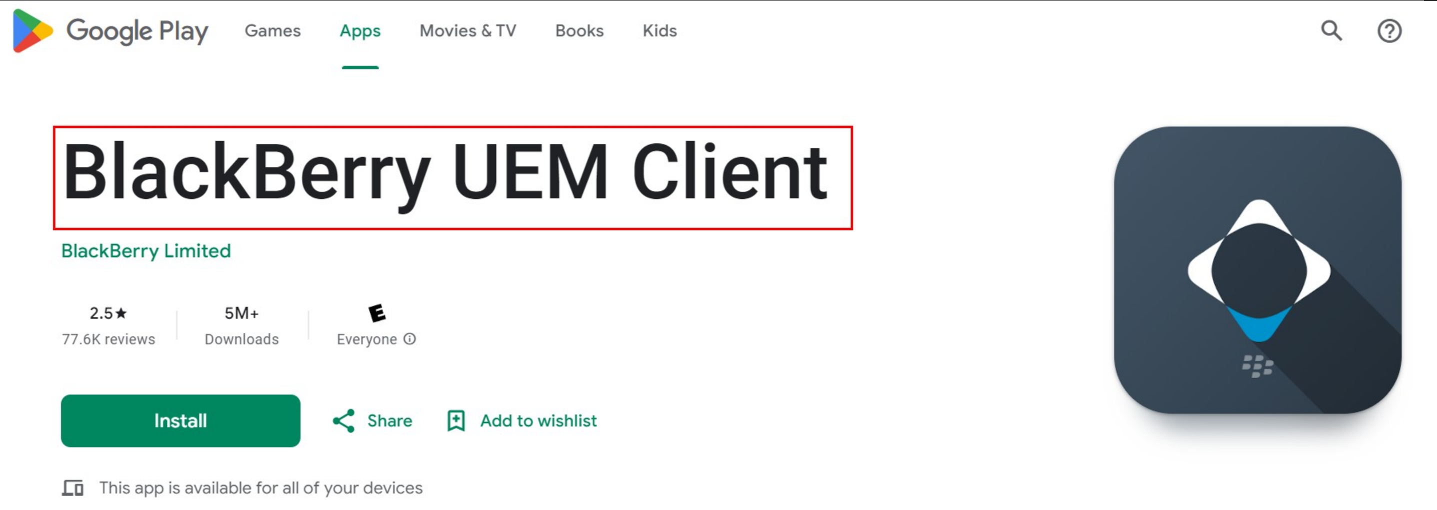
- The SafetyNet or Google Play Integrity attestation failure option is a compliance profile setting for Android devices and BlackBerry Dynamics apps that allows you to specify the actions that occur if devices or apps do not pass attestation. To set this option, navigate to **Policies and profiles > Compliance > Android** tab.
- BlackBerry UEM uses the Play Integrity API with UEM Client versions that support it to provide additional protection from application tampering. Play Integrity will replace SafetyNet based on the migration schedule that is determined by Google and will continue to use SafetyNet for earlier UEM Client versions. For more information about migrating from SafetyNet, see the information from Google.

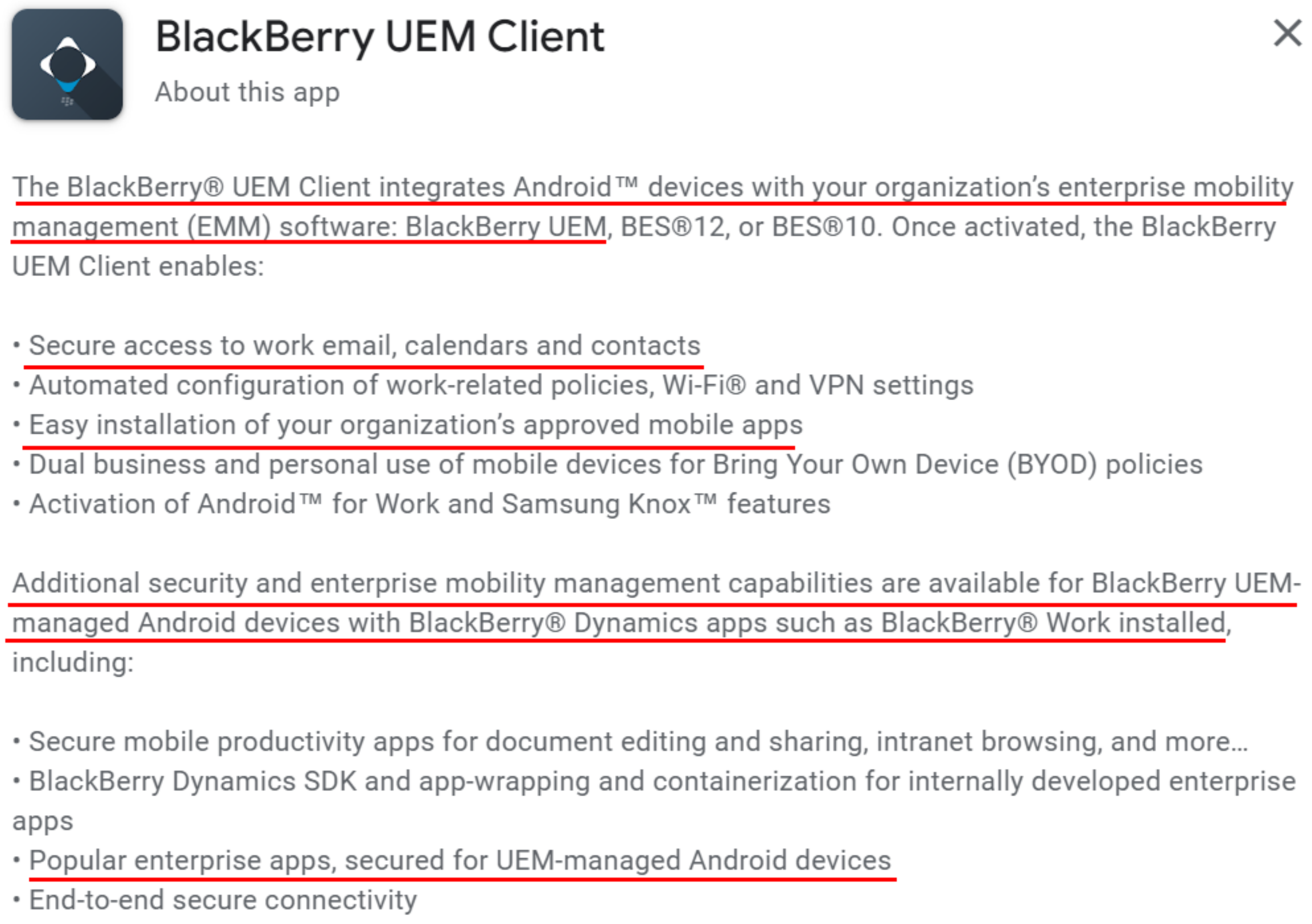
	https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_18/administration/device-features-it-policies/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/Considerations-for-configuring-attestation-for-Android-devices-and-apps-using-SafetyNet  **What is Play Integrity?** The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions. You can use the Play Integrity API to obtain integrity verdicts about your app and device, which can help you respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access. https://developer.android.com/codelabs/add-play-integrity#0  1. The server creates a globally unique value in a way that malicious users cannot predict. For example, a cryptographically-secure random number 128 bits or larger. 2. Your app calls the Play Integrity API, and sets the nonce field to the unique value received by your app server. 3. Your app sends the signed result of the Play Integrity API to your server. 4. Your server verifies that the nonce field in the signed result matches the unique value it previously generated, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1

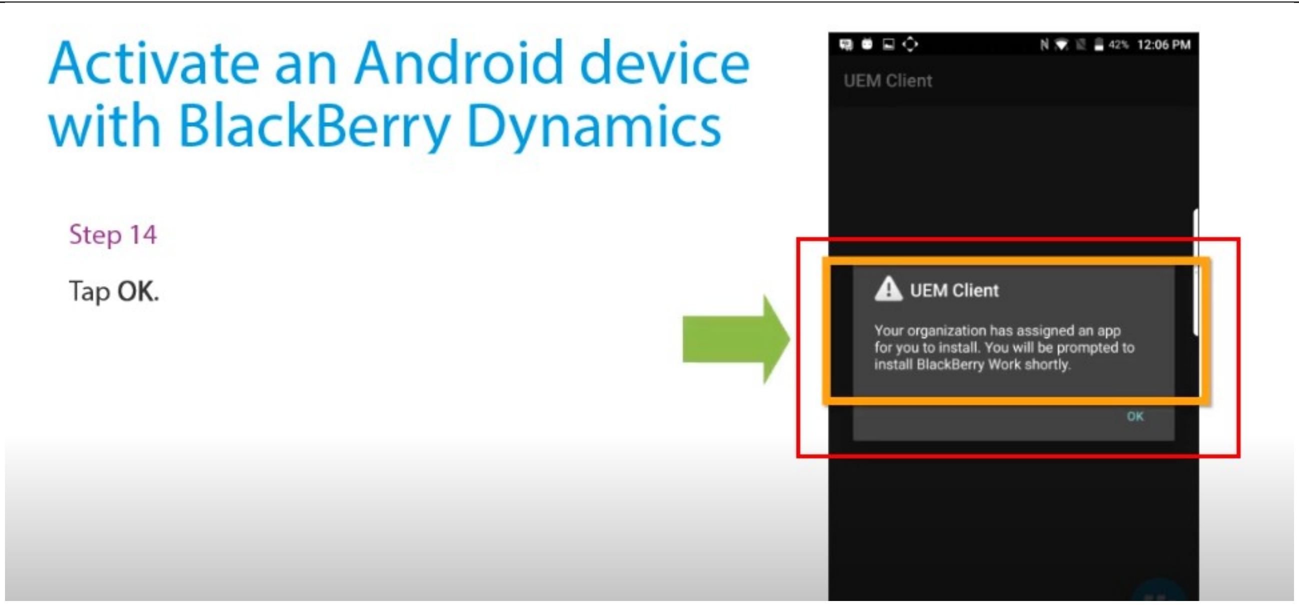
	 Your app server Your app Google Play Generate a globally-unique value Integrated Software Computer Program Send the unique value Software Application logic module Call `setNonce()` with the received value Call the Play Integrity API API returns signed result Send the signed result Operational assurance logic module Verify that the `nonce` field matches the unique value Your app server Your app Google Play https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1
wherein the software application logic module is operable to provide a plurality of sub-procedures to provide computing functions associated with the	The accused instrumentality discloses wherein the software application logic module (e.g., application algorithms, etc.) is operable to provide a plurality of sub-procedures (e.g., algorithms for various app functions, etc.) to provide computing functions (e.g., program instructions, encryption, etc.) associated with the software application logic module (e.g., application algorithms, etc.) execution of the integrated software computer program (e.g., a work application along with attestation API, etc.) and as a part of the integrated software computer program (e.g., a work application along with

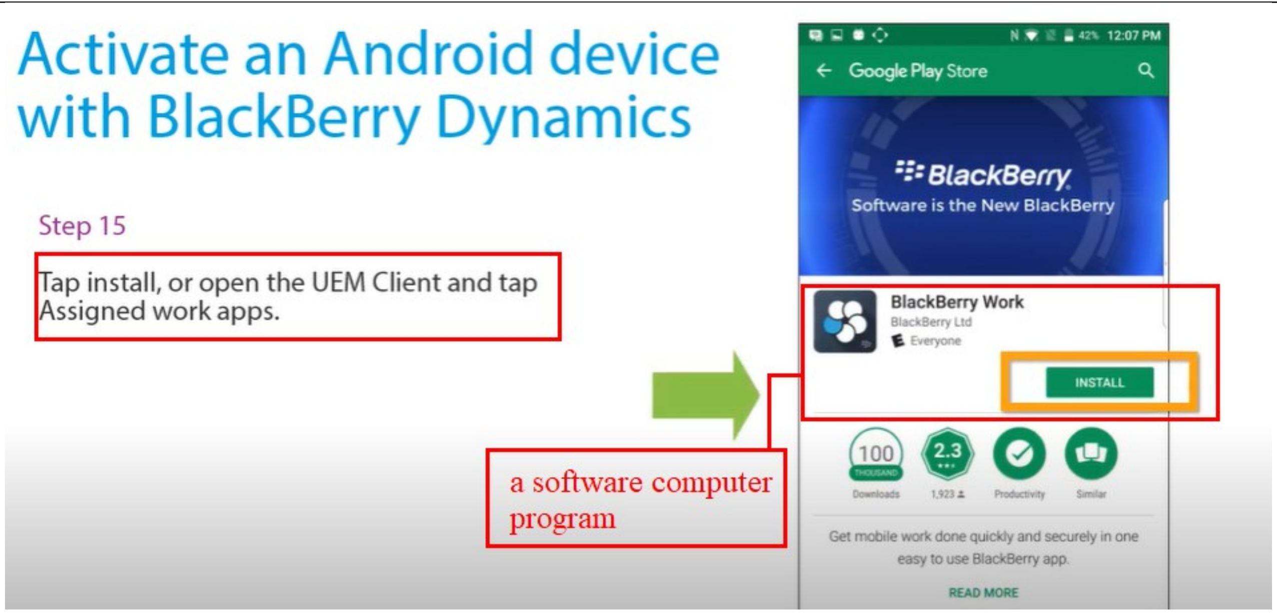
software application logic module execution of the integrated software computer program and as a part of the integrated software computer program; **Col 28: lines 14-20** *The resulting system is such wherein the intermixture is further comprised of at least one of: obfuscation, encryption, replication, adding dummy code, addition of redundant control, renaming of variables,* ***<u>splitting a procedure into multiple sub-procedure,</u>*** *dictionary transformation, compilation, interpretation, cryptographic transformation, digital signing, and scrambling.*	attestation API, etc.). As shown below, the accused instrumentality is a Unified Endpoint Management (UEM) solution that allows enterprises to manage and control devices linked to employees. It allows administrators to add work applications that are permitted for use by employees on their managed devices. Also, the administrator can set the available functions for a specific work application. For example, an application such as E-mail (e.g., an integrated computer software program) includes various algorithms and processes to compose and send email to a recipient.  BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps Managing apps In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

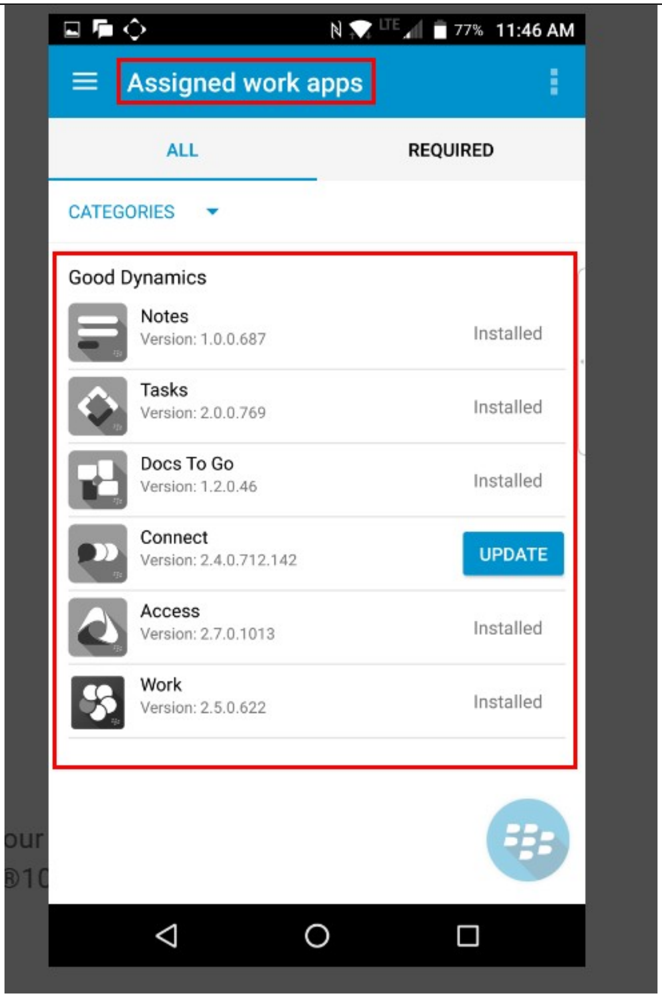
	 Add public and internal apps to UEM. — Add apps to the apps list so that you can assign them to users' devices. You can add public apps, such as those from the App Store and Google Play store. You can also add internal apps which you upload the source files for. You can specify app configurations, which allow you to preconfigure certain app settings before you assign apps to users. By preconfiguring app settings, you can make it easier for users to download, set up, and use the apps. For example, many apps require users to type a URL, an email address, or other information before they can use the app. By adding an app configuration, you can configure some of these settings in advance. You can create multiple app configurations for an app with different settings for different purposes, and rank the configurations. If an app is assigned to a user more than once with different app configurations, the app with the highest rank is applied. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps  You use the BlackBerry UEM Client to activate your device for work. When you activate your device, the device is associated with BlackBerry UEM and is granted access to work data and the productivity apps that your administrator assigned to your device. Your administrator determines the degree of protection for your device based on your role and assigns IT policies and profiles to make sure the appropriate device features are available to you and to secure work data on your device. https://docs.blackberry.com/en/endpoint-management/blackberry-uem-client/latest#:~:text=You%20use%20the%20BlackBerry%20UEM,work%20data%20on%20your%20device.

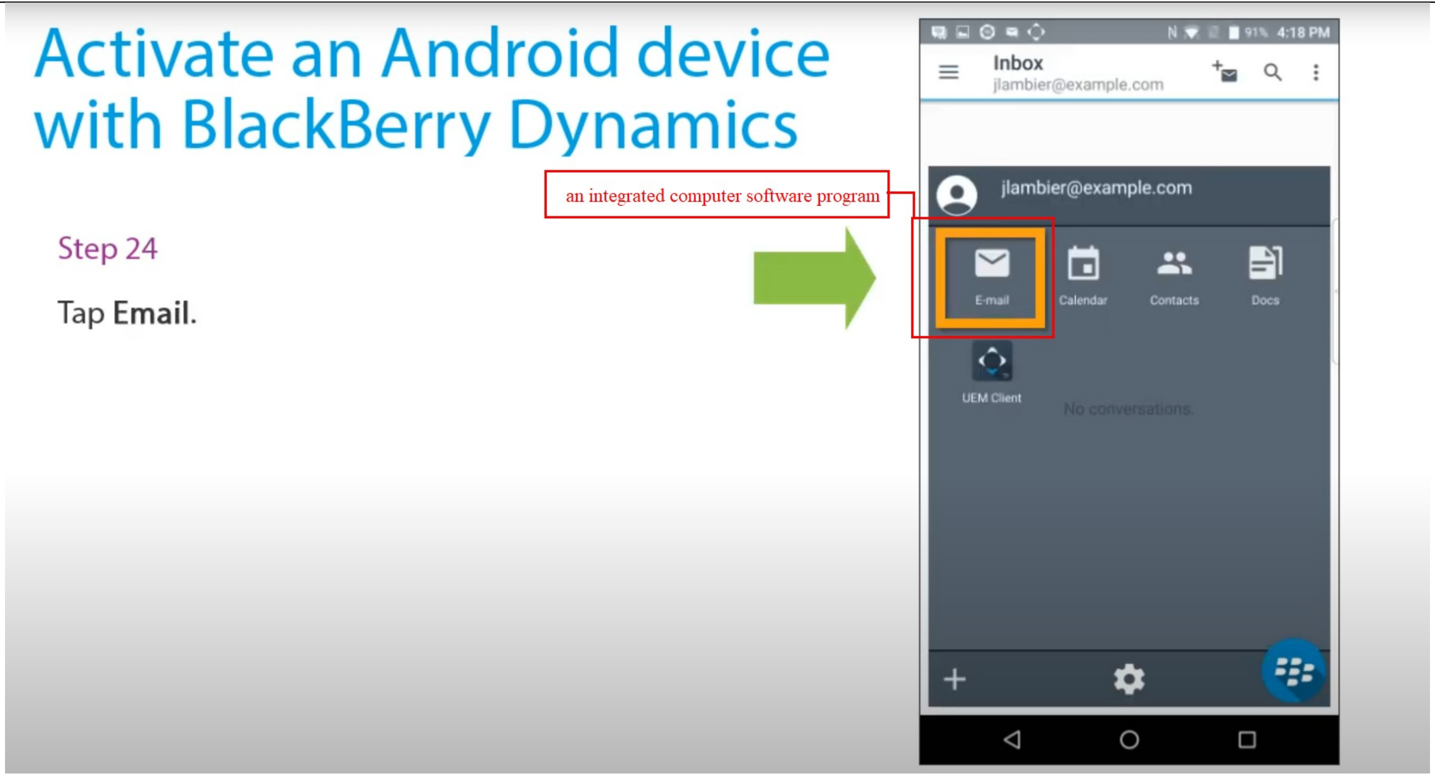
	 Google Play Games Apps Movies & TV Books Kids BlackBerry UEM Client BlackBerry Limited 2.5★ 77.6K reviews · 5M+ Downloads · Everyone Install Share Add to wishlist This app is available for all of your devices https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

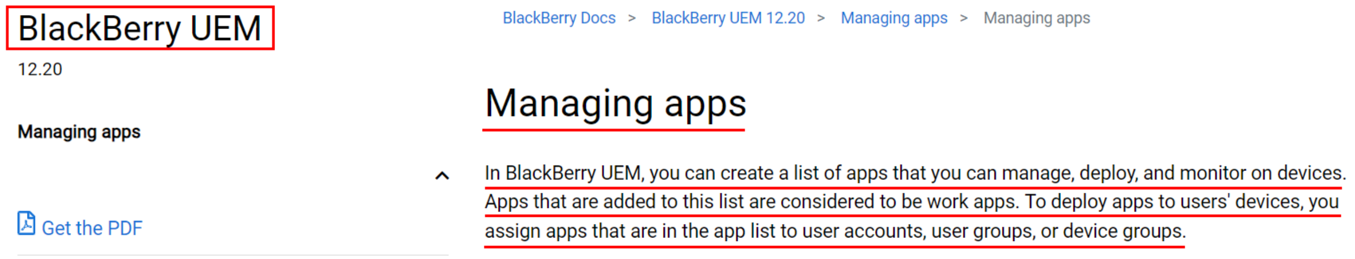
	 BlackBerry UEM Client About this app The BlackBerry® UEM Client integrates Android™ devices with your organization's enterprise mobility management (EMM) software: BlackBerry UEM, BES®12, or BES®10. Once activated, the BlackBerry UEM Client enables: • Secure access to work email, calendars and contacts • Automated configuration of work-related policies, Wi-Fi® and VPN settings • Easy installation of your organization's approved mobile apps • Dual business and personal use of mobile devices for Bring Your Own Device (BYOD) policies • Activation of Android™ for Work and Samsung Knox™ features Additional security and enterprise mobility management capabilities are available for BlackBerry UEM-managed Android devices with BlackBerry® Dynamics apps such as BlackBerry® Work installed, including: • Secure mobile productivity apps for document editing and sharing, intranet browsing, and more... • BlackBerry Dynamics SDK and app-wrapping and containerization for internally developed enterprise apps • Popular enterprise apps, secured for UEM-managed Android devices • End-to-end secure connectivity https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

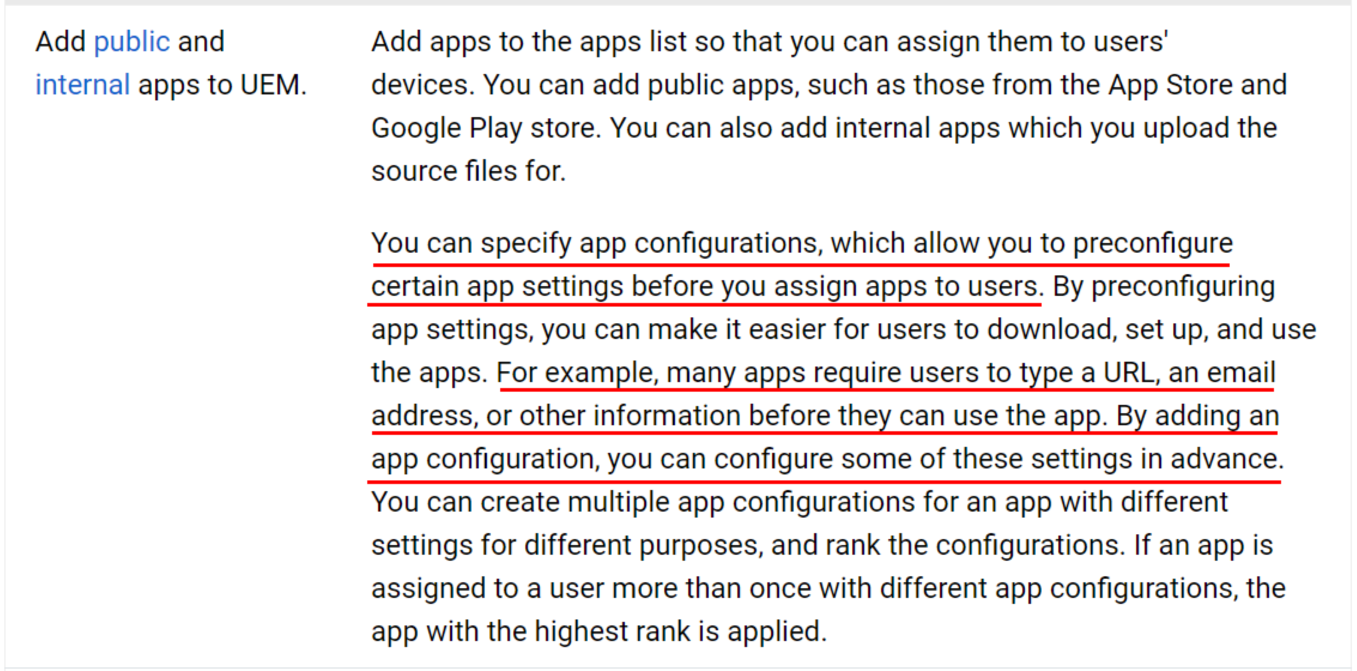
	 Activate an Android device with BlackBerry Dynamics Step 14 Tap **OK.** UEM Client Your organization has assigned an app for you to install. You will be prompted to install BlackBerry Work shortly. OK https://www.youtube.com/watch?v=zRrI1TpLVXM

	 Activate an Android device with BlackBerry Dynamics Step 15 Tap install, or open the UEM Client and tap Assigned work apps. a software computer program Google Play Store BlackBerry Software is the New BlackBerry BlackBerry Work BlackBerry Ltd Everyone INSTALL 100 THOUSAND Downloads 2.3 1,923 Productivity Similar Get mobile work done quickly and securely in one easy to use BlackBerry app. READ MORE https://www.youtube.com/watch?v=zRrI1TpLVXM

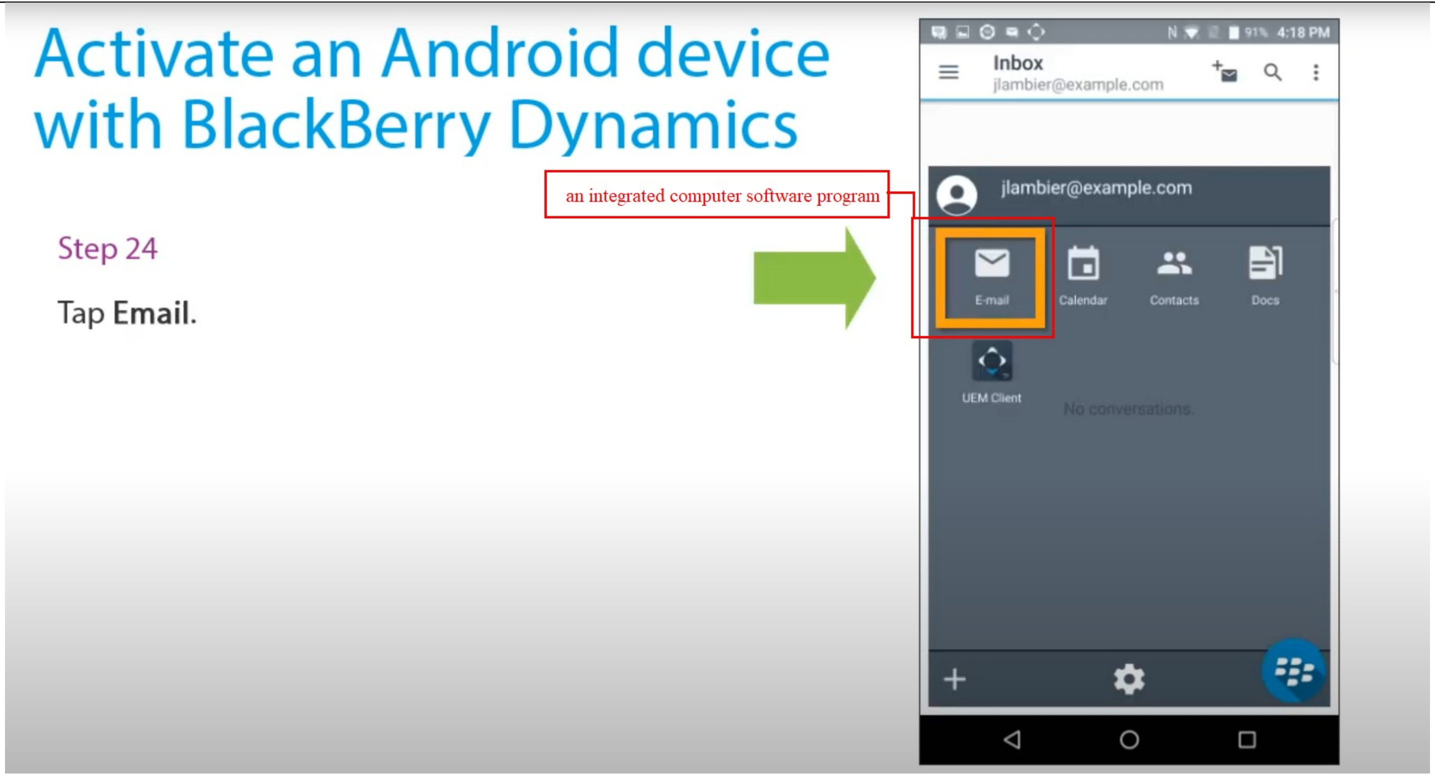
	 77% 11:46 AM Assigned work apps ALL REQUIRED CATEGORIES Good Dynamics Notes Version: 1.0.0.687 Installed Tasks Version: 2.0.0.769 Installed Docs To Go Version: 1.2.0.46 Installed Connect Version: 2.4.0.712.142 UPDATE Access Version: 2.7.0.1013 Installed Work Version: 2.5.0.622 Installed https://play.google.com/store/apps/details?id=com.rim.mobilefusion.client&hl=en_US

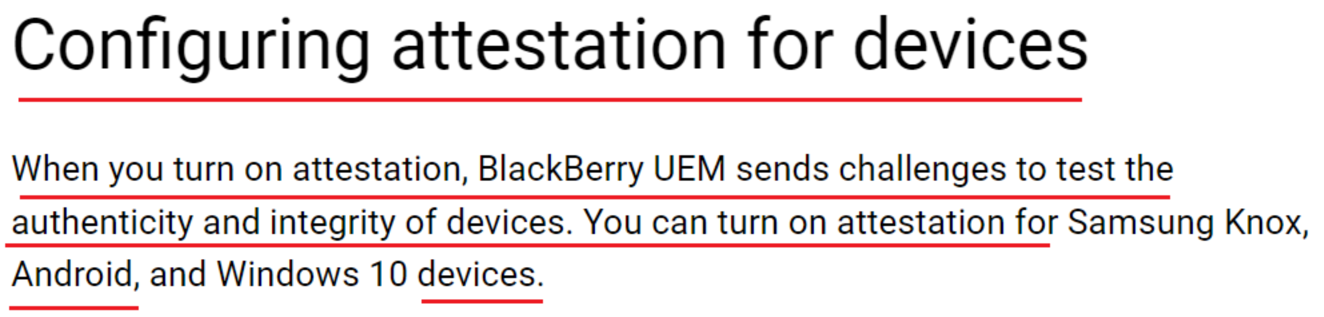
	 Activate an Android device with BlackBerry Dynamics Step 24 Tap **Email.** an integrated computer software program Inbox jlambier@example.com jlambier@example.com E-mail Calendar Contacts Docs UEM Client No conversations. https://www.youtube.com/watch?v=zRrI1TpLVXM
wherein the operational assurance logic provides for generation of unique security tags securely associated with the execution of the integrated software computer program;	The accused instrumentality discloses wherein the operational assurance logic (e.g., Attestation API, etc.) provides for generation of unique security tags (e.g., a nonce, signed token, etc.) securely associated with the execution of the integrated software computer program (e.g., a work application along with attestation API, etc.). As shown below, the accused instrumentality is a Unified Endpoint Management (UEM) solution that allows enterprises to manage and control devices linked to employees. It allows administrators to add work applications that are permitted for use by employees on their managed devices.

	Also, the accused instrumentality utilizes device attestation for verifying device & app integrity. When execution of a work application (e.g., a software computer program) running on an employee device requires integrity verification, it utilizes device attestation feature for the verification. It utilizes an attestation API (e.g., an operation assurance logic module) such as Play Integrity API for an android device of an employee. The work app sends a request to the attestation API, it sends a signed token (e.g., unique security tag, etc.) back to the app. The app forwards the signed token to a server for verification.  BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps **Managing apps** In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps



	Add public and internal apps to UEM. Add apps to the apps list so that you can assign them to users' devices. You can add public apps, such as those from the App Store and Google Play store. You can also add internal apps which you upload the source files for. You can specify app configurations, which allow you to preconfigure certain app settings before you assign apps to users. By preconfiguring app settings, you can make it easier for users to download, set up, and use the apps. For example, many apps require users to type a URL, an email address, or other information before they can use the app. By adding an app configuration, you can configure some of these settings in advance. You can create multiple app configurations for an app with different settings for different purposes, and rank the configurations. If an app is assigned to a user more than once with different app configurations, the app with the highest rank is applied. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

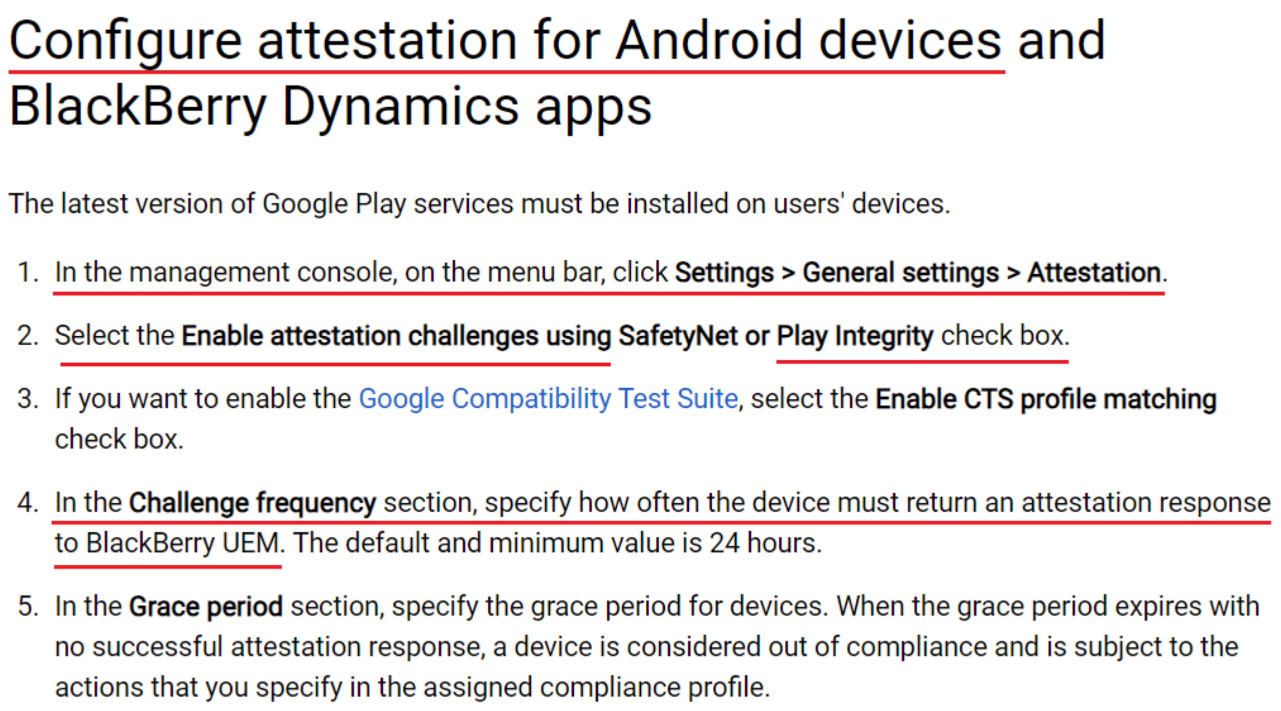
	 Activate an Android device with BlackBerry Dynamics Step 24 Tap **Email.** an integrated computer software program Inbox jlambier@example.com jlambier@example.com E-mail Calendar Contacts Docs UEM Client No conversations. 4:18 PM https://www.youtube.com/watch?v=zRrI1TpLVXM

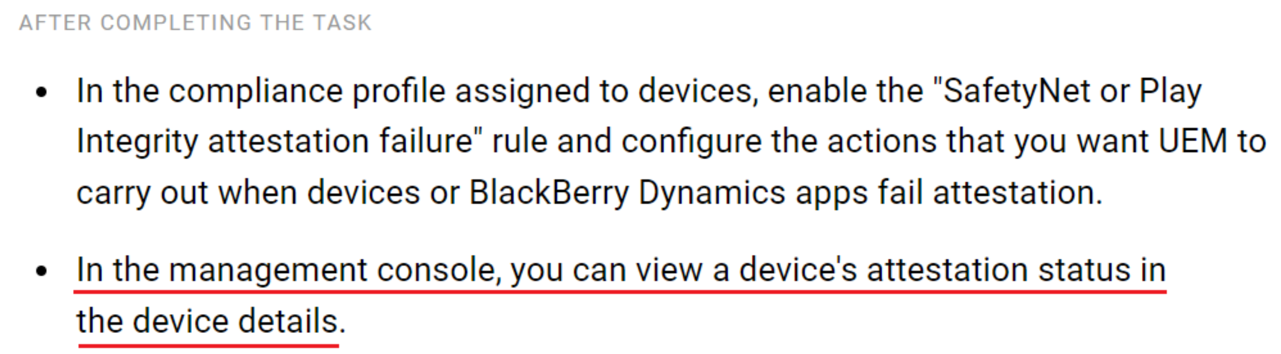
<table>
<tr><td></td><td>

Configuring attestation for devices

When you turn on attestation, BlackBerry UEM sends challenges to test the authenticity and integrity of devices. You can turn on attestation for Samsung Knox, Android, and Windows 10 devices.

https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation</td></tr>
</table>

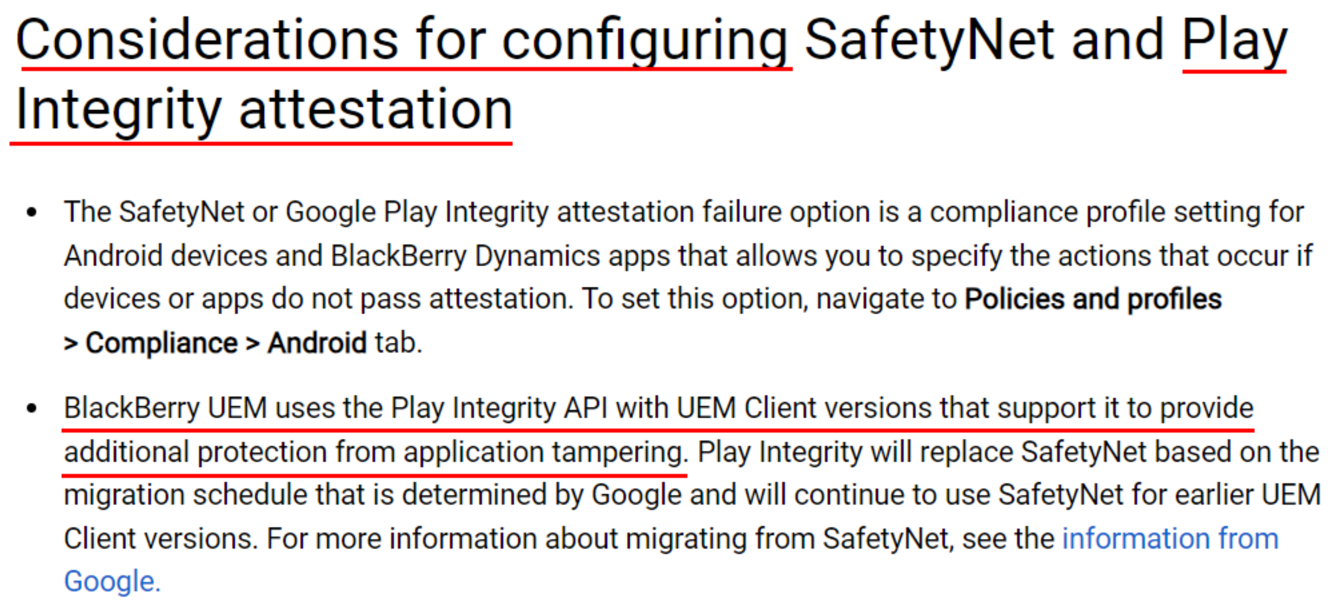
	 Configure attestation for Android devices and BlackBerry Dynamics apps The latest version of Google Play services must be installed on users' devices. 1. In the management console, on the menu bar, click **Settings > General settings > Attestation**. 2. Select the **Enable attestation challenges using SafetyNet or Play Integrity** check box. 3. If you want to enable the Google Compatibility Test Suite, select the **Enable CTS profile matching** check box. 4. In the **Challenge frequency** section, specify how often the device must return an attestation response to BlackBerry UEM. The default and minimum value is 24 hours. 5. In the **Grace period** section, specify the grace period for devices. When the grace period expires with no successful attestation response, a device is considered out of compliance and is subject to the actions that you specify in the assigned compliance profile. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841



AFTER COMPLETING THE TASK

- In the compliance profile assigned to devices, enable the "SafetyNet or Play Integrity attestation failure" rule and configure the actions that you want UEM to carry out when devices or BlackBerry Dynamics apps fail attestation.
- In the management console, you can view a device's attestation status in the device details.

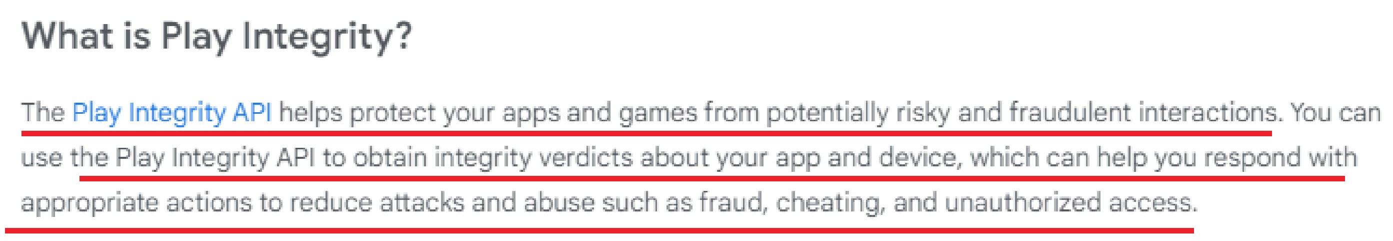
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841



Considerations for configuring SafetyNet and Play Integrity attestation

- The SafetyNet or Google Play Integrity attestation failure option is a compliance profile setting for Android devices and BlackBerry Dynamics apps that allows you to specify the actions that occur if devices or apps do not pass attestation. To set this option, navigate to **Policies and profiles > Compliance > Android** tab.
- BlackBerry UEM uses the Play Integrity API with UEM Client versions that support it to provide additional protection from application tampering. Play Integrity will replace SafetyNet based on the migration schedule that is determined by Google and will continue to use SafetyNet for earlier UEM Client versions. For more information about migrating from SafetyNet, see the information from Google.

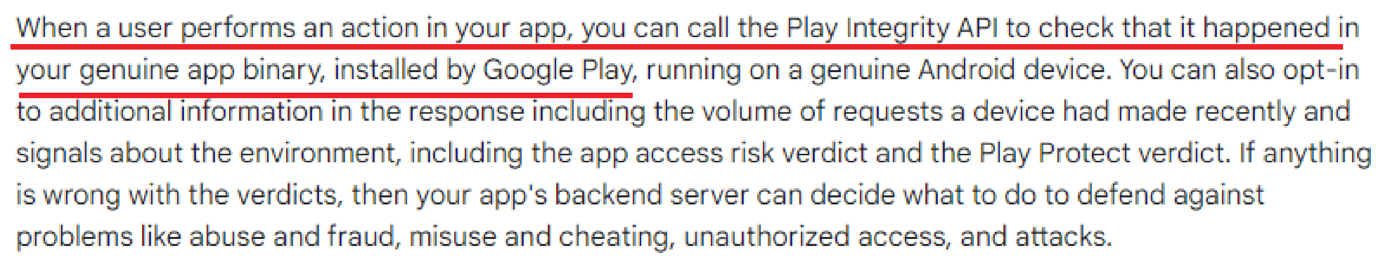
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_18/administration/device-features-it-policies/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/Considerations-for-configuring-attestation-for-Android-devices-and-apps-using-SafetyNet



What is Play Integrity?

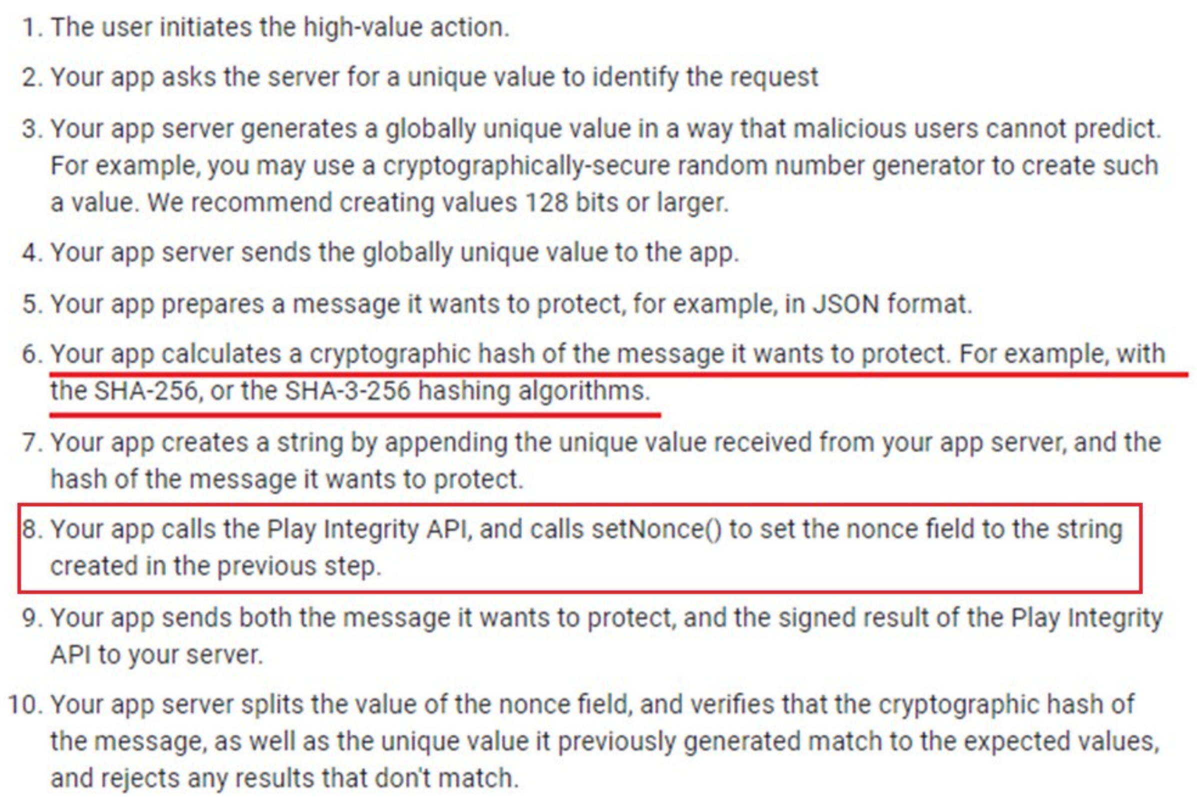
The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions. You can use the Play Integrity API to obtain integrity verdicts about your app and device, which can help you respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access.

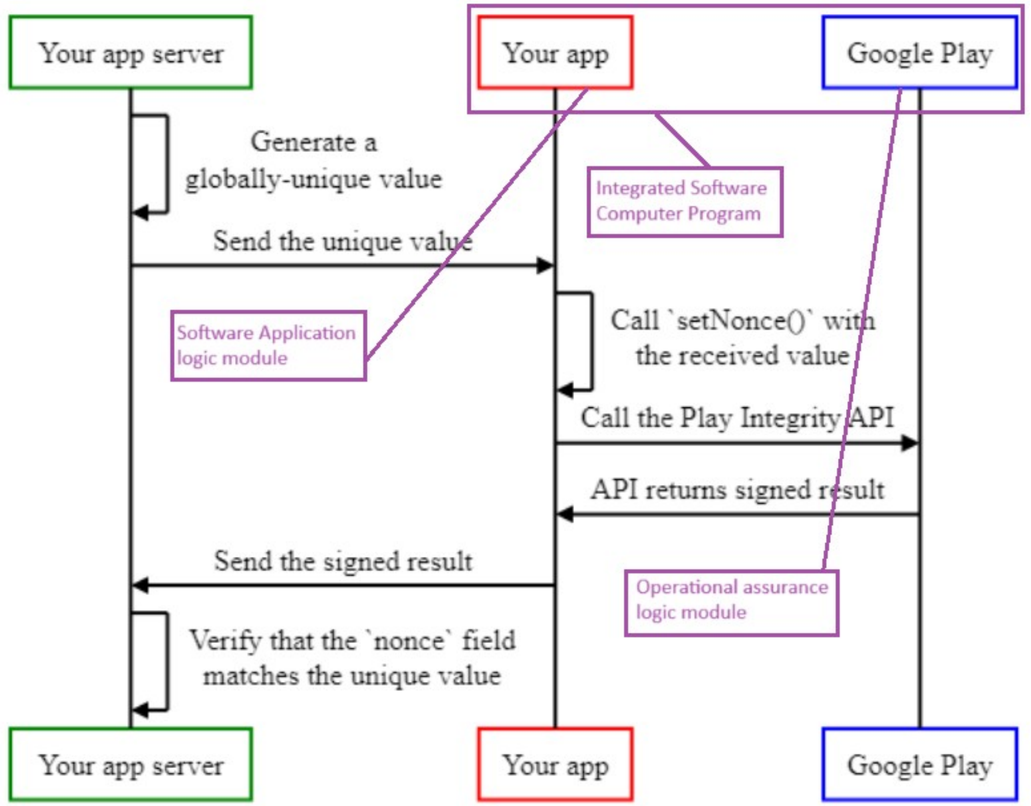
https://developer.android.com/codelabs/add-play-integrity#0

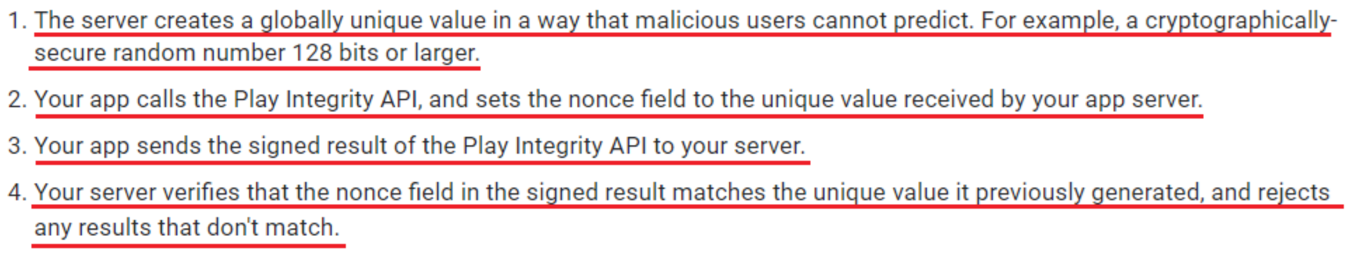


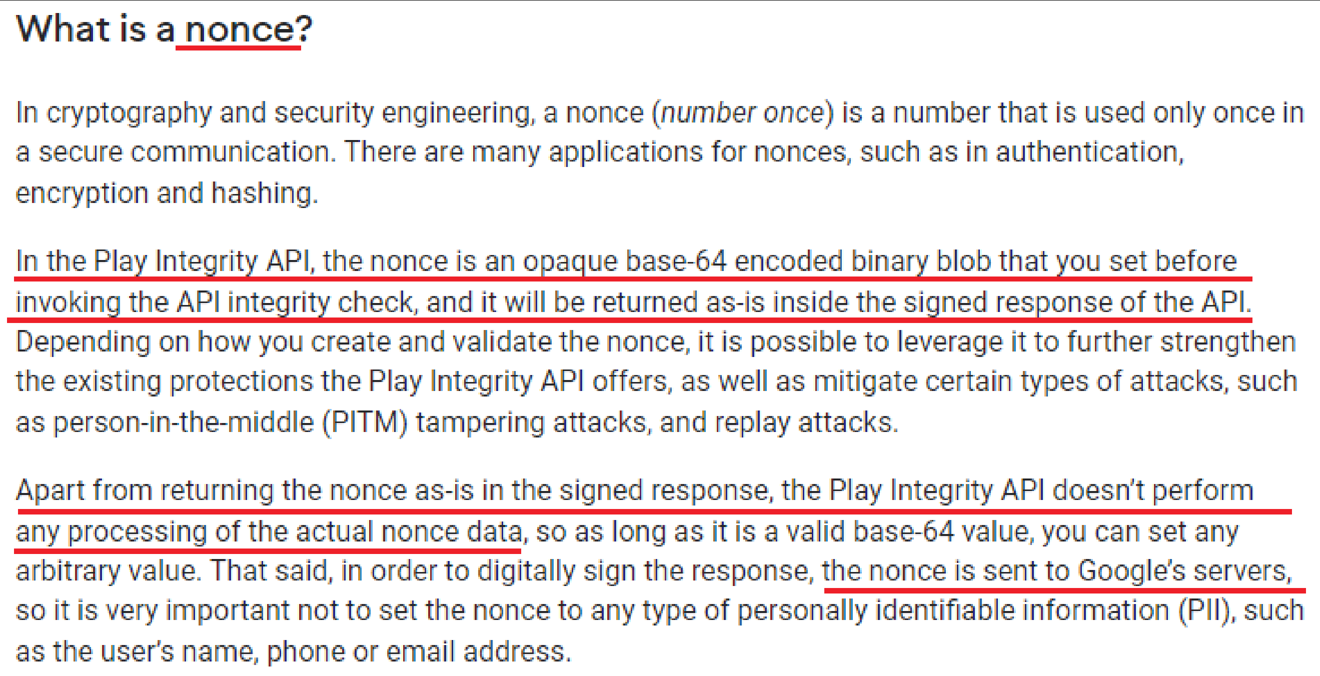
When a user performs an action in your app, you can call the Play Integrity API to check that it happened in your genuine app binary, installed by Google Play, running on a genuine Android device. You can also opt-in to additional information in the response including the volume of requests a device had made recently and signals about the environment, including the app access risk verdict and the Play Protect verdict. If anything is wrong with the verdicts, then your app's backend server can decide what to do to defend against problems like abuse and fraud, misuse and cheating, unauthorized access, and attacks.

https://developer.android.com/google/play/integrity/overview#:~:text=When%20a%20user%20performs%20an,%2C%20unauthorized%20access%2C%20and%20attacks.

	 1. The user initiates the high-value action. 2. Your app asks the server for a unique value to identify the request 3. Your app server generates a globally unique value in a way that malicious users cannot predict. For example, you may use a cryptographically-secure random number generator to create such a value. We recommend creating values 128 bits or larger. 4. Your app server sends the globally unique value to the app. 5. Your app prepares a message it wants to protect, for example, in JSON format. 6. Your app calculates a cryptographic hash of the message it wants to protect. For example, with the SHA-256, or the SHA-3-256 hashing algorithms. 7. Your app creates a string by appending the unique value received from your app server, and the hash of the message it wants to protect. 8. Your app calls the Play Integrity API, and calls setNonce() to set the nonce field to the string created in the previous step. 9. Your app sends both the message it wants to protect, and the signed result of the Play Integrity API to your server. 10. Your app server splits the value of the nonce field, and verifies that the cryptographic hash of the message, as well as the unique value it previously generated match to the expected values, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

	 Your app server Your app Google Play Generate a globally-unique value Integrated Software Computer Program Send the unique value Software Application logic module Call `setNonce()` with the received value Call the Play Integrity API API returns signed result Send the signed result Operational assurance logic module Verify that the `nonce` field matches the unique value Your app server Your app Google Play https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1

	 1. The server creates a globally unique value in a way that malicious users cannot predict. For example, a cryptographically-secure random number 128 bits or larger. 2. Your app calls the Play Integrity API, and sets the nonce field to the unique value received by your app server. 3. Your app sends the signed result of the Play Integrity API to your server. 4. Your server verifies that the nonce field in the signed result matches the unique value it previously generated, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1 • A token generated by the Play Integrity API https://developer.android.com/codelabs/add-play-integrity#2



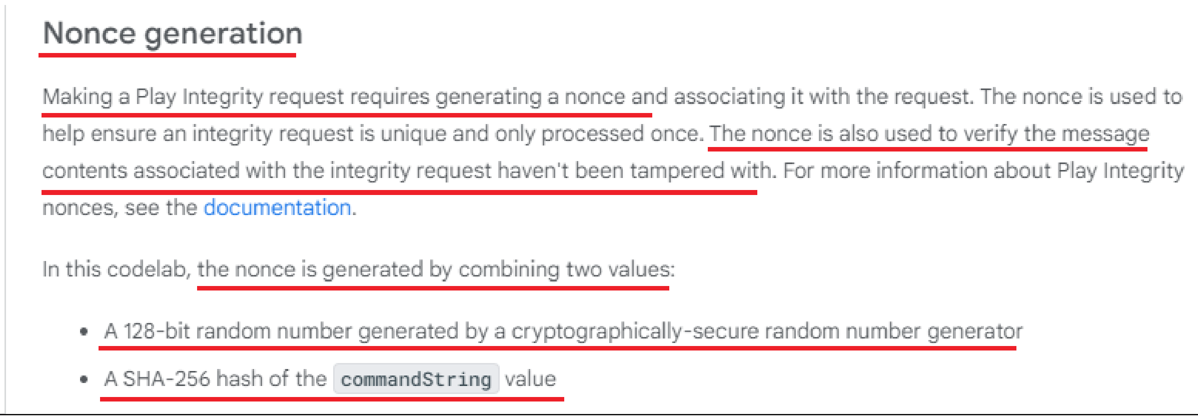
What is a nonce?

In cryptography and security engineering, a nonce (*number once*) is a number that is used only once in a secure communication. There are many applications for nonces, such as in authentication, encryption and hashing.

In the Play Integrity API, the nonce is an opaque base-64 encoded binary blob that you set before invoking the API integrity check, and it will be returned as-is inside the signed response of the API. Depending on how you create and validate the nonce, it is possible to leverage it to further strengthen the existing protections the Play Integrity API offers, as well as mitigate certain types of attacks, such as person-in-the-middle (PITM) tampering attacks, and replay attacks.

Apart from returning the nonce as-is in the signed response, the Play Integrity API doesn't perform any processing of the actual nonce data, so as long as it is a valid base-64 value, you can set any arbitrary value. That said, in order to digitally sign the response, the nonce is sent to Google's servers, so it is very important not to set the nonce to any type of personally identifiable information (PII), such as the user's name, phone or email address.

https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

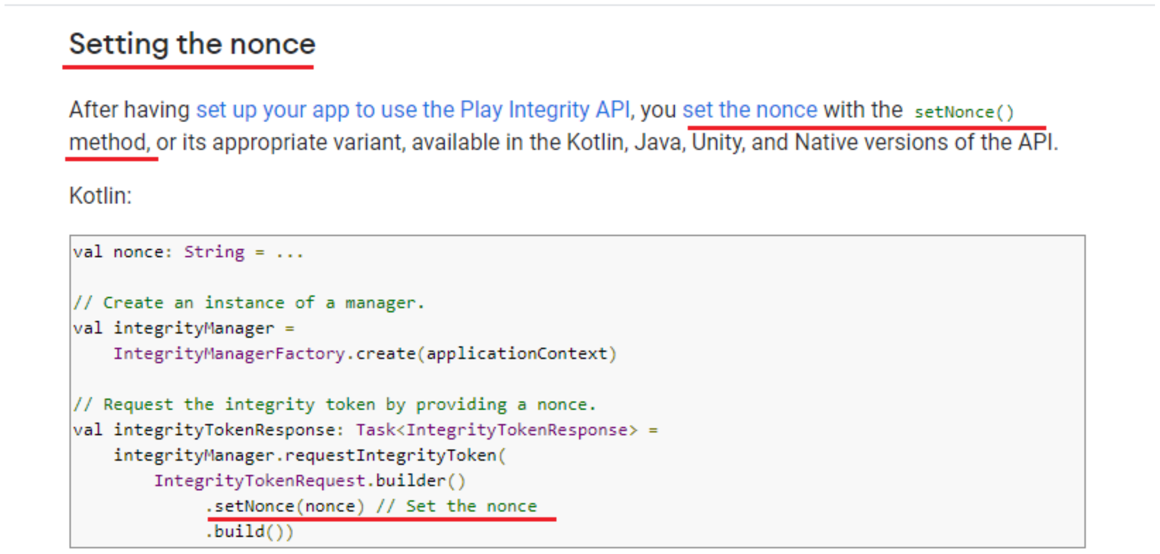


Nonce generation

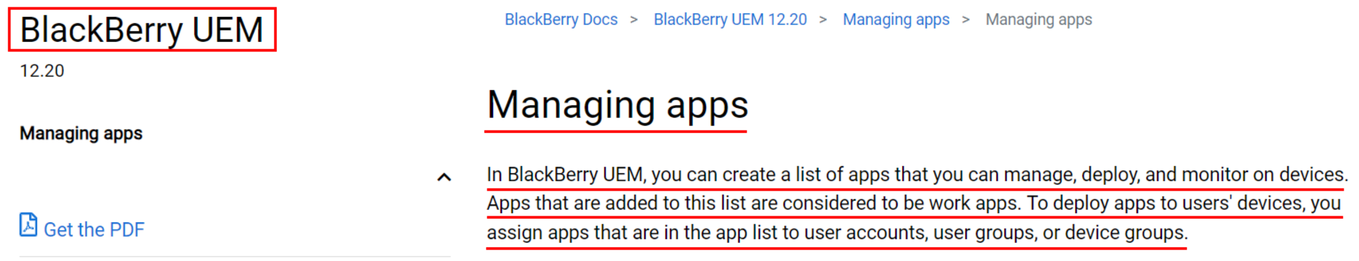
Making a Play Integrity request requires generating a nonce and associating it with the request. The nonce is used to help ensure an integrity request is unique and only processed once. The nonce is also used to verify the message contents associated with the integrity request haven't been tampered with. For more information about Play Integrity nonces, see the documentation.

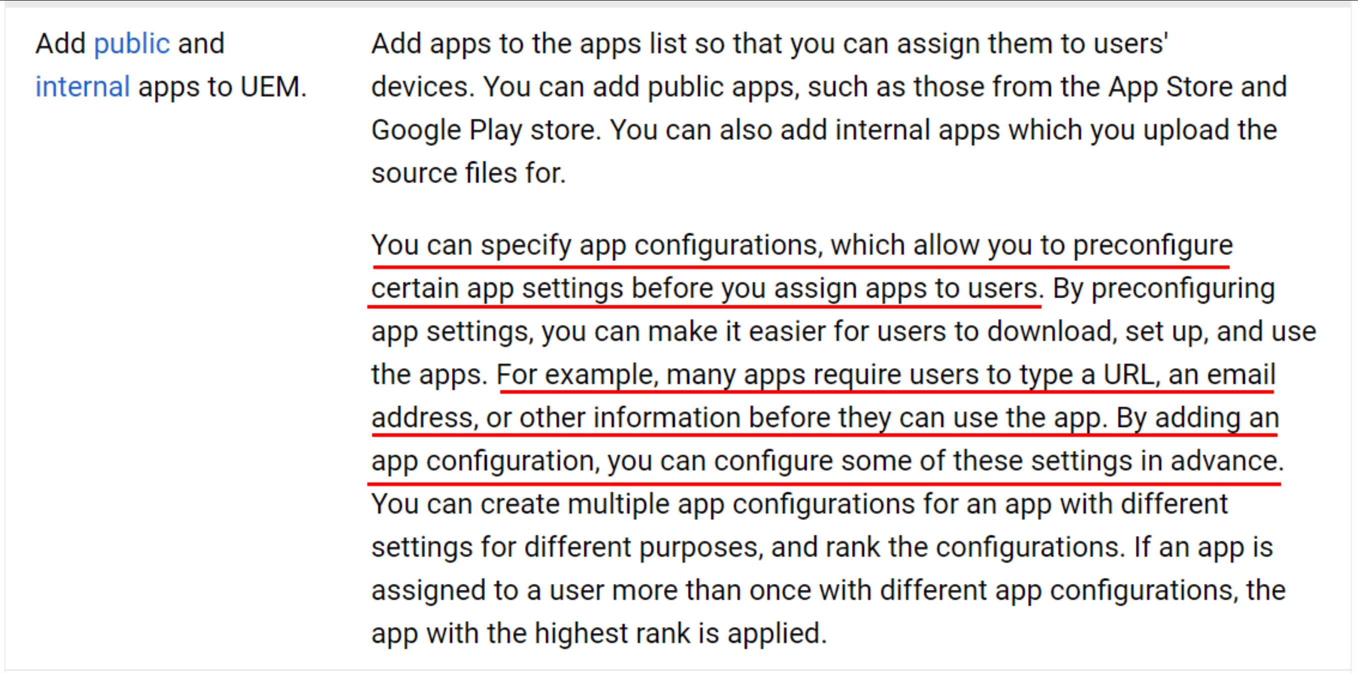
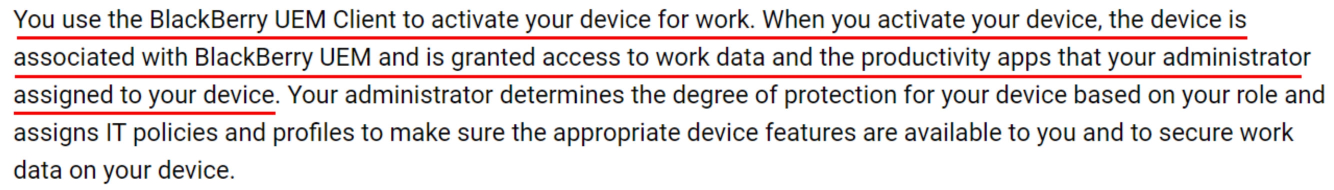
In this codelab, the nonce is generated by combining two values:

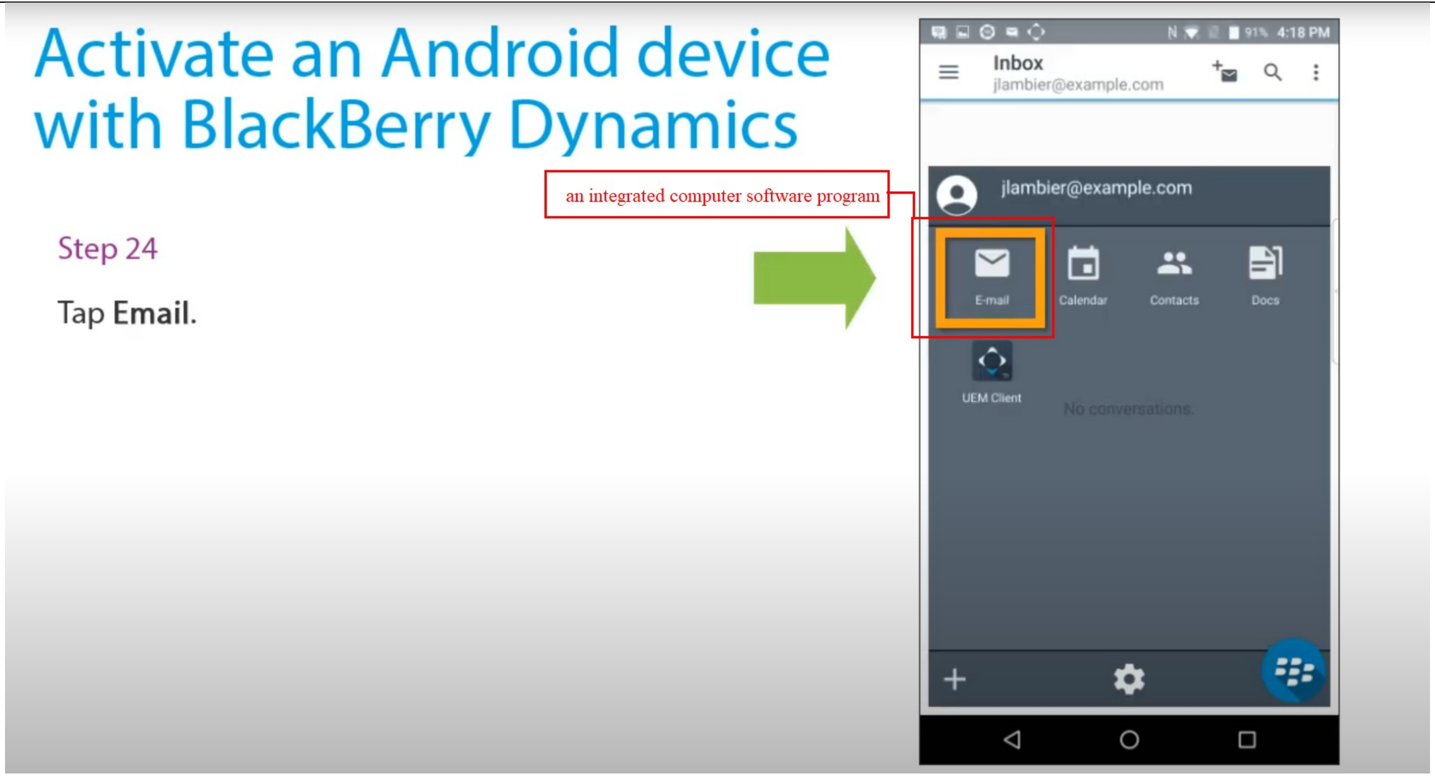
- A 128-bit random number generated by a cryptographically-secure random number generator
- A SHA-256 hash of the `commandString` value

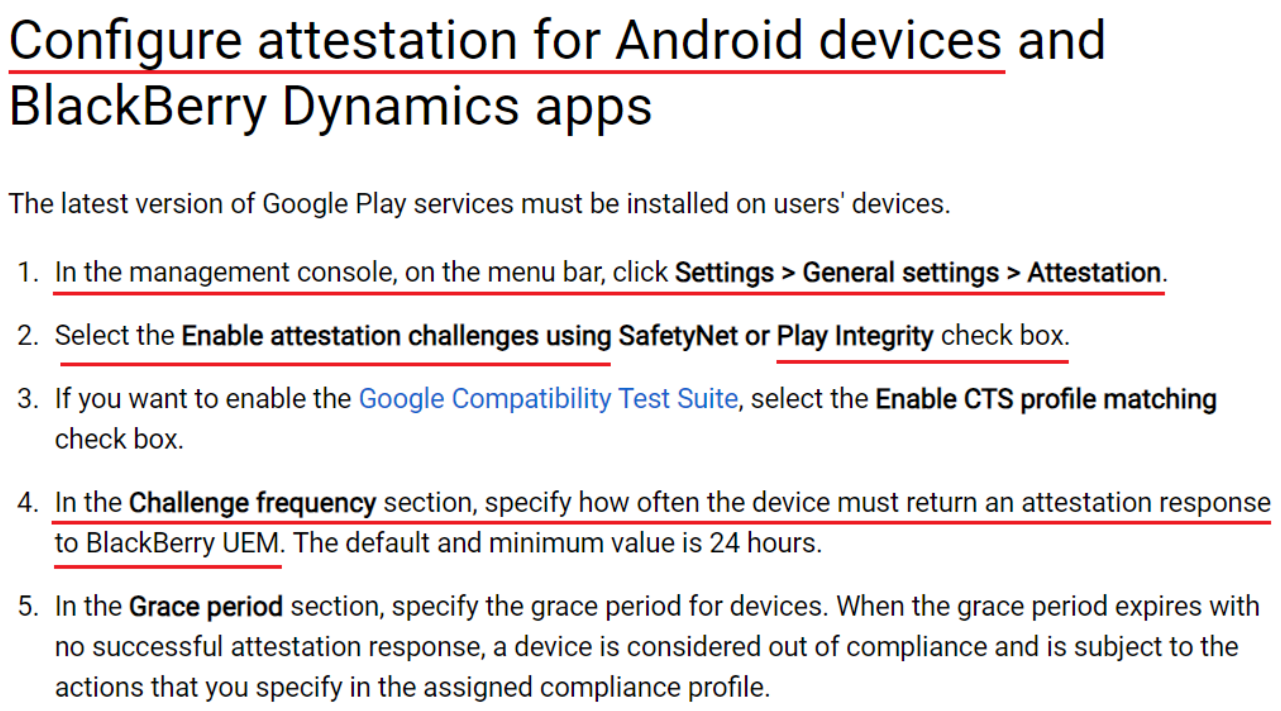
	https://developer.android.com/codelabs/add-play-integrity#2  Setting the nonce After having set up your app to use the Play Integrity API, you set the nonce with the `setNonce()` method, or its appropriate variant, available in the Kotlin, Java, Unity, and Native versions of the API. Kotlin:
	```
	val nonce: String = ...

	// Create an instance of a manager.
	val integrityManager =
	    IntegrityManagerFactory.create(applicationContext)

	// Request the integrity token by providing a nonce.
	val integrityTokenResponse: Task<IntegrityTokenResponse> =
	    integrityManager.requestIntegrityToken(
	        IntegrityTokenRequest.builder()
	            .setNonce(nonce) // Set the nonce
	            .build())
	```
	 https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

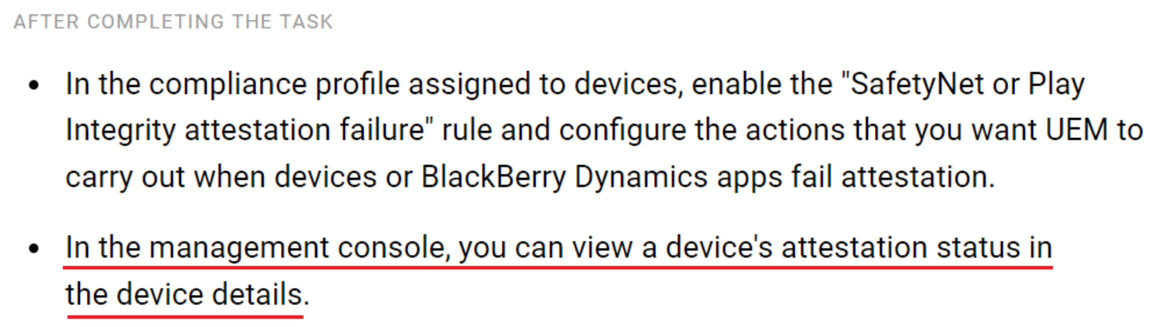
executing the integrated software computer program to provide generation of the unique software keys by the operation assurance logic when the integrated software computer program as executed is unaltered, to provide thr validating	The accused instrumentality discloses executing the integrated software computer program (e.g., a work application along with attestation API, etc.) to provide generation of the unique software keys (e.g., a nonce, signed token, etc.) by the operation assurance logic (e.g., Attestation API, etc.) when the integrated software computer program (e.g., a work application along with attestation API, etc.) as executed is unaltered, to provide the validating that the integrated software computer program (e.g., a work application along with attestation API, etc.) that is being executed was not altered and that there was no tampering of the integrated software computer program (e.g., a work application along with attestation API, etc.). As shown below, the accused instrumentality is a Unified Endpoint Management (UEM)

that the integrated software computer program that is being executed was not altered and that there was no tampering of the integrated software computer program;	solution that allows enterprises to manage and control devices linked to employees. It allows administrators to add work applications that are permitted for use by employees on their managed devices. Also, the accused instrumentality utilizes device attestation for verifying device & app integrity. When execution of a work application (e.g., a software computer program) running on an employee device requires integrity verification, it utilizes device attestation feature for the verification. It utilizes an attestation API (e.g., an operation assurance logic module) such as Play Integrity API for an android device of an employee. The work app sends a request to the attestation API, it sends a signed token (e.g., unique security tag, etc.) back to the app. The app forwards the signed token to a server for verification. The server verifies whether the signed token matches its previously generated unique value. Thus, if the values match, it determines the valid app is executing a program on a valid device.  BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps Managing apps In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

	 Add public and internal apps to UEM. — Add apps to the apps list so that you can assign them to users' devices. You can add public apps, such as those from the App Store and Google Play store. You can also add internal apps which you upload the source files for. You can specify app configurations, which allow you to preconfigure certain app settings before you assign apps to users. By preconfiguring app settings, you can make it easier for users to download, set up, and use the apps. For example, many apps require users to type a URL, an email address, or other information before they can use the app. By adding an app configuration, you can configure some of these settings in advance. You can create multiple app configurations for an app with different settings for different purposes, and rank the configurations. If an app is assigned to a user more than once with different app configurations, the app with the highest rank is applied. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps  You use the BlackBerry UEM Client to activate your device for work. When you activate your device, the device is associated with BlackBerry UEM and is granted access to work data and the productivity apps that your administrator assigned to your device. Your administrator determines the degree of protection for your device based on your role and assigns IT policies and profiles to make sure the appropriate device features are available to you and to secure work data on your device. https://docs.blackberry.com/en/endpoint-management/blackberry-uem-client/latest#:~:text=You%20use%20the%20BlackBerry%20UEM,work%20data%20on%20your%20device.

	 Activate an Android device with BlackBerry Dynamics Step 24 Tap **Email.** an integrated computer software program Inbox jlambier@example.com jlambier@example.com E-mail Calendar Contacts Docs UEM Client No conversations. https://www.youtube.com/watch?v=zRrI1TpLVXM

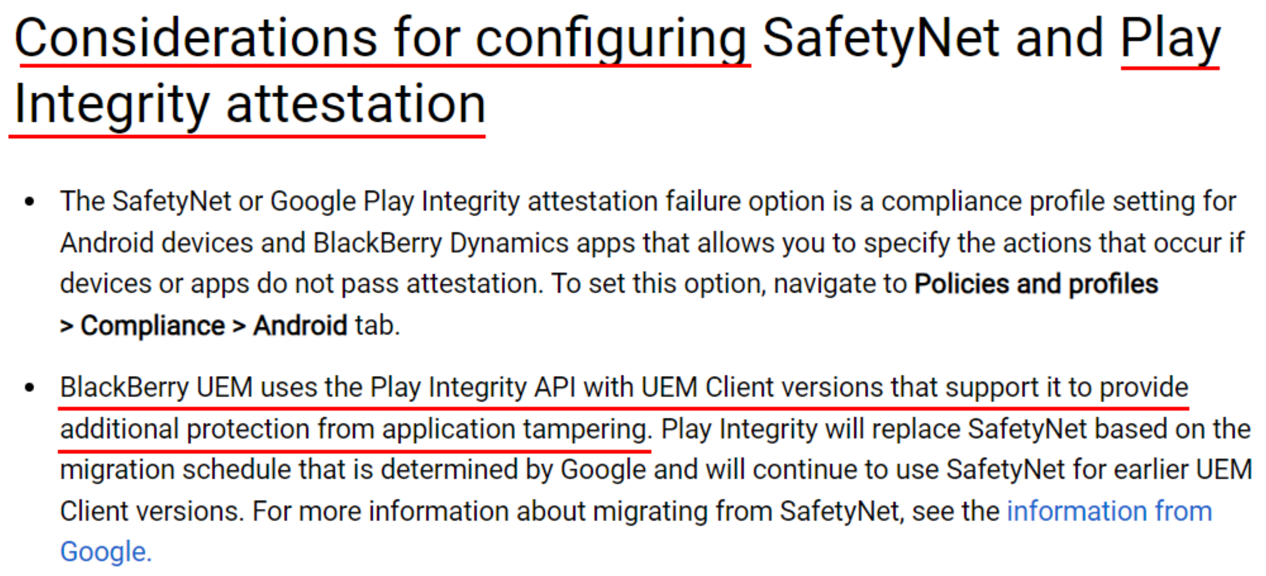
	 **Configure attestation for Android devices and BlackBerry Dynamics apps** The latest version of Google Play services must be installed on users' devices. 1. In the management console, on the menu bar, click **Settings > General settings > Attestation**. 2. Select the **Enable attestation challenges using SafetyNet or Play Integrity** check box. 3. If you want to enable the Google Compatibility Test Suite, select the **Enable CTS profile matching** check box. 4. In the **Challenge frequency** section, specify how often the device must return an attestation response to BlackBerry UEM. The default and minimum value is 24 hours. 5. In the **Grace period** section, specify the grace period for devices. When the grace period expires with no successful attestation response, a device is considered out of compliance and is subject to the actions that you specify in the assigned compliance profile. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841



AFTER COMPLETING THE TASK

- In the compliance profile assigned to devices, enable the "SafetyNet or Play Integrity attestation failure" rule and configure the actions that you want UEM to carry out when devices or BlackBerry Dynamics apps fail attestation.
- In the management console, you can view a device's attestation status in the device details.

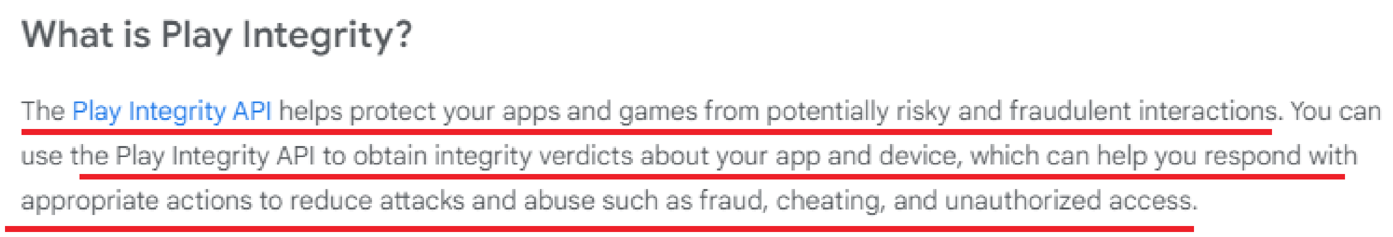
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841



Considerations for configuring SafetyNet and Play Integrity attestation

- The SafetyNet or Google Play Integrity attestation failure option is a compliance profile setting for Android devices and BlackBerry Dynamics apps that allows you to specify the actions that occur if devices or apps do not pass attestation. To set this option, navigate to **Policies and profiles > Compliance > Android** tab.
- BlackBerry UEM uses the Play Integrity API with UEM Client versions that support it to provide additional protection from application tampering. Play Integrity will replace SafetyNet based on the migration schedule that is determined by Google and will continue to use SafetyNet for earlier UEM Client versions. For more information about migrating from SafetyNet, see the information from Google.

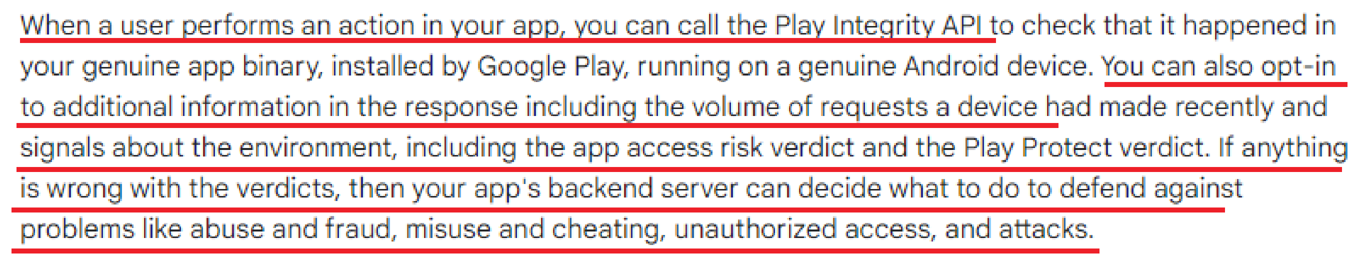
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_18/administration/device-features-it-policies/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/Considerations-for-configuring-attestation-for-Android-devices-and-apps-using-SafetyNet



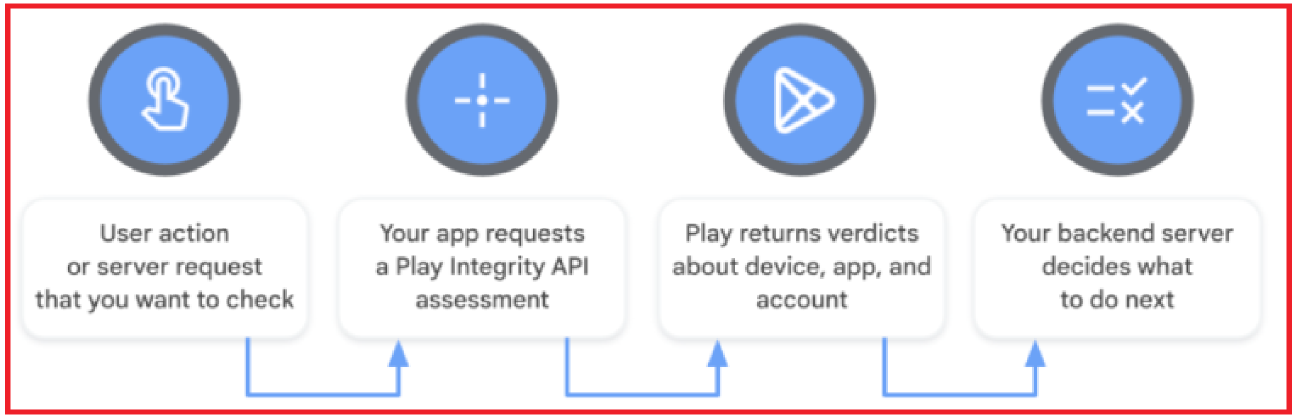
What is Play Integrity?

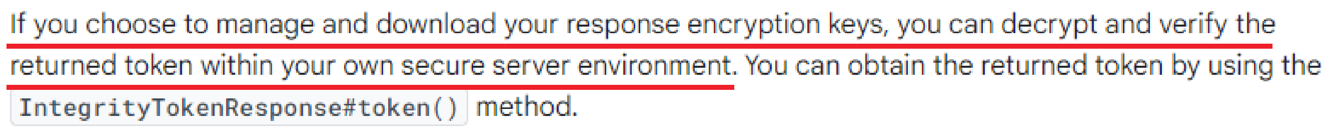
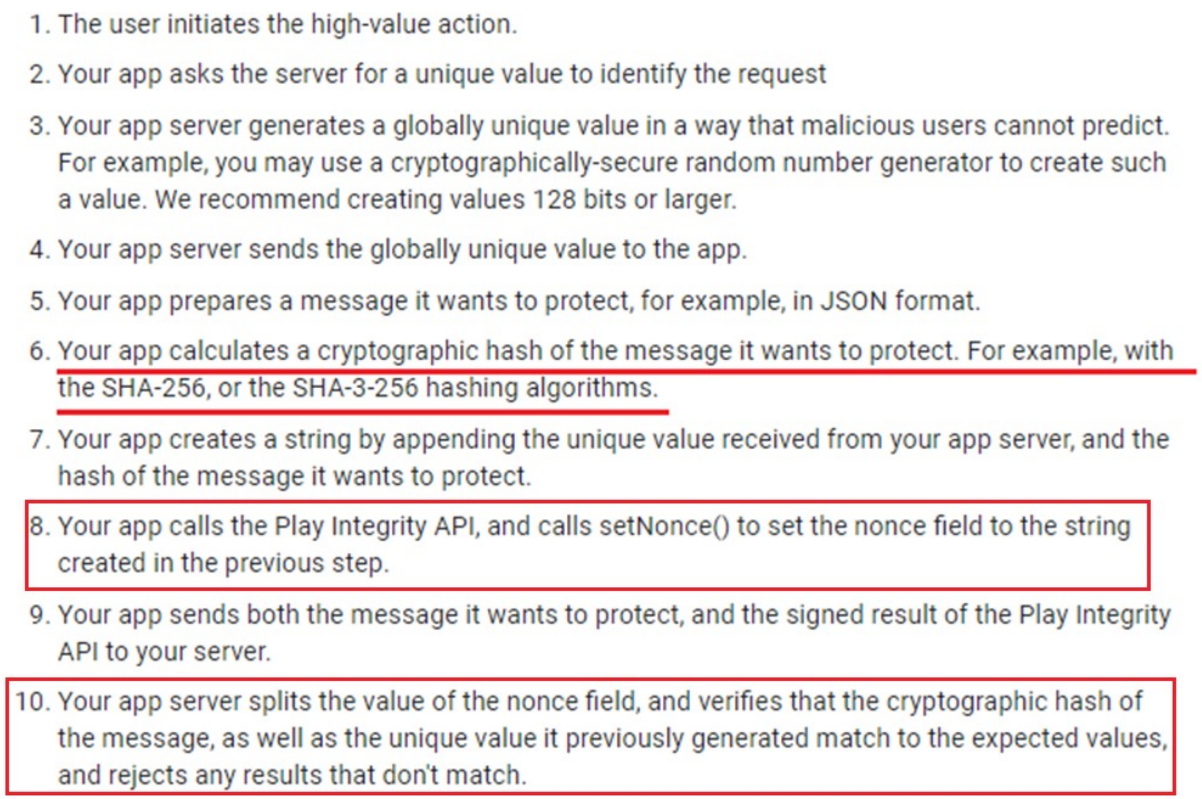
The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions. You can use the Play Integrity API to obtain integrity verdicts about your app and device, which can help you respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access.

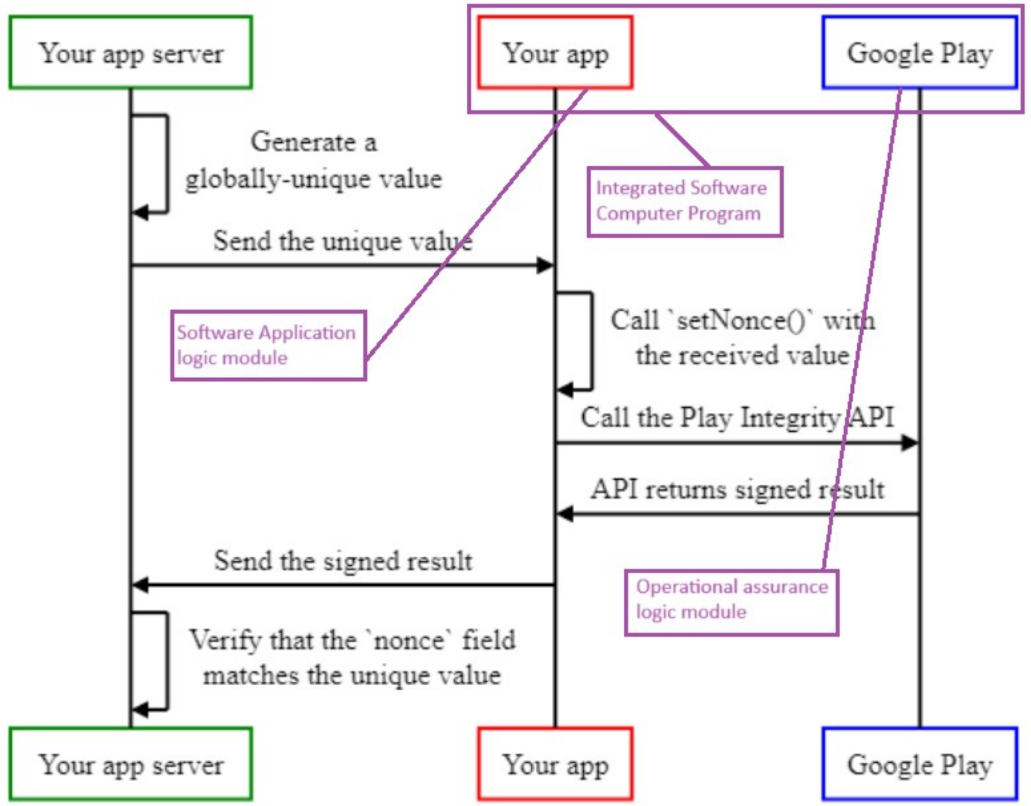
https://developer.android.com/codelabs/add-play-integrity#0

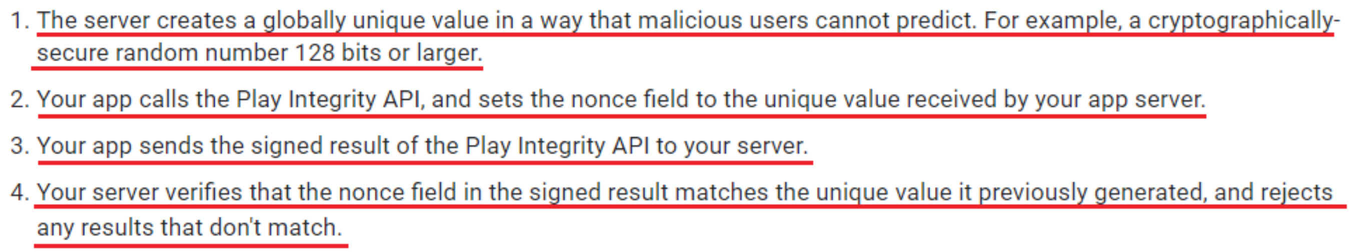


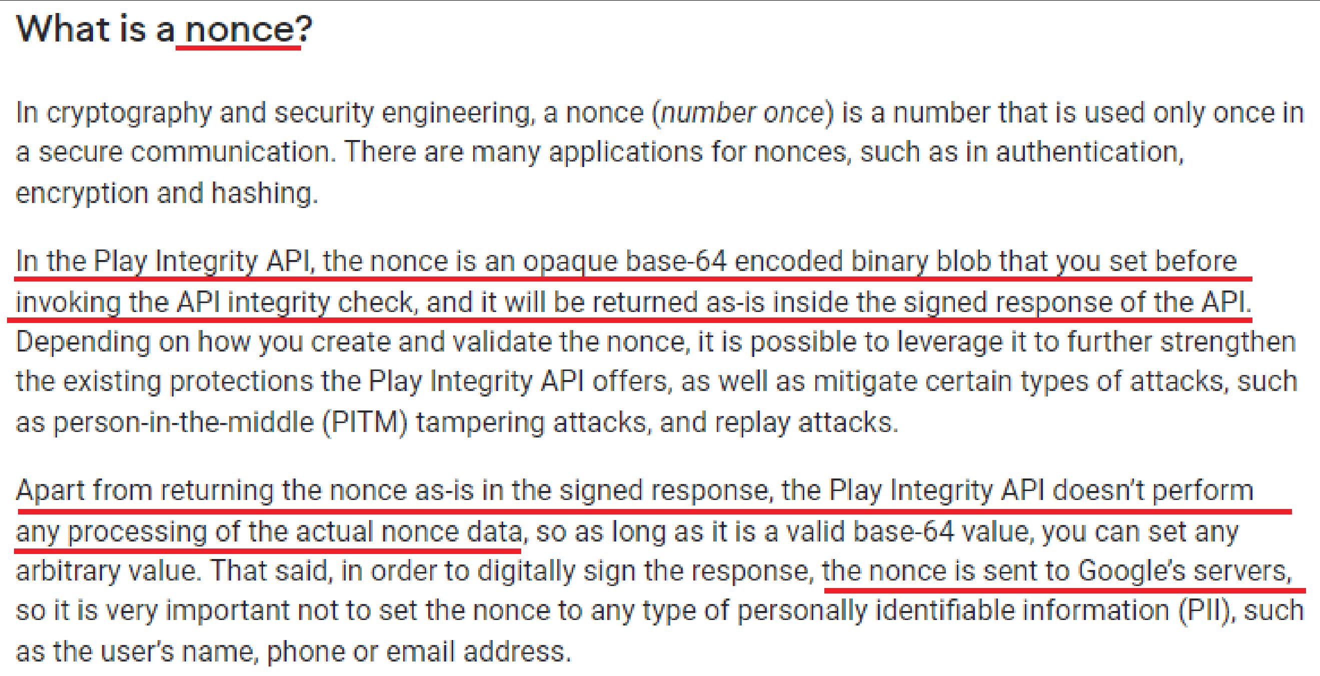
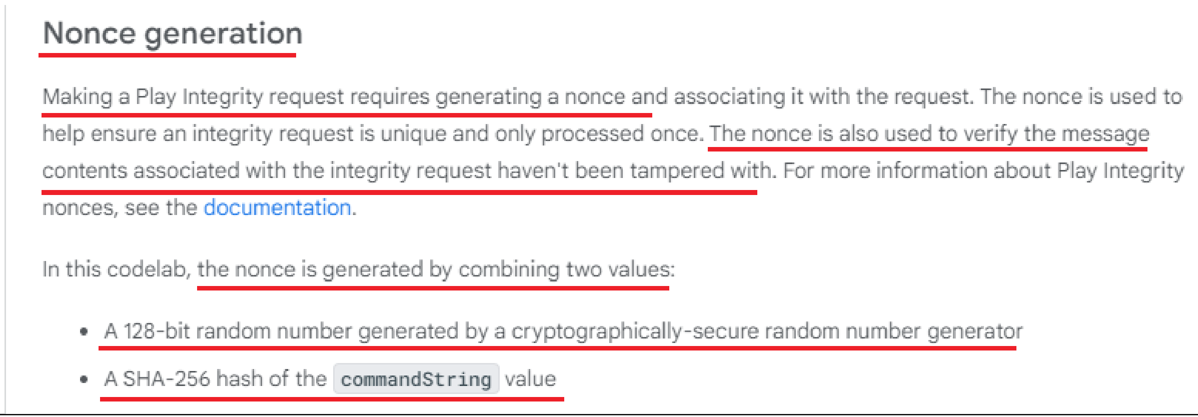
When a user performs an action in your app, you can call the Play Integrity API to check that it happened in your genuine app binary, installed by Google Play, running on a genuine Android device. You can also opt-in to additional information in the response including the volume of requests a device had made recently and signals about the environment, including the app access risk verdict and the Play Protect verdict. If anything is wrong with the verdicts, then your app's backend server can decide what to do to defend against problems like abuse and fraud, misuse and cheating, unauthorized access, and attacks.

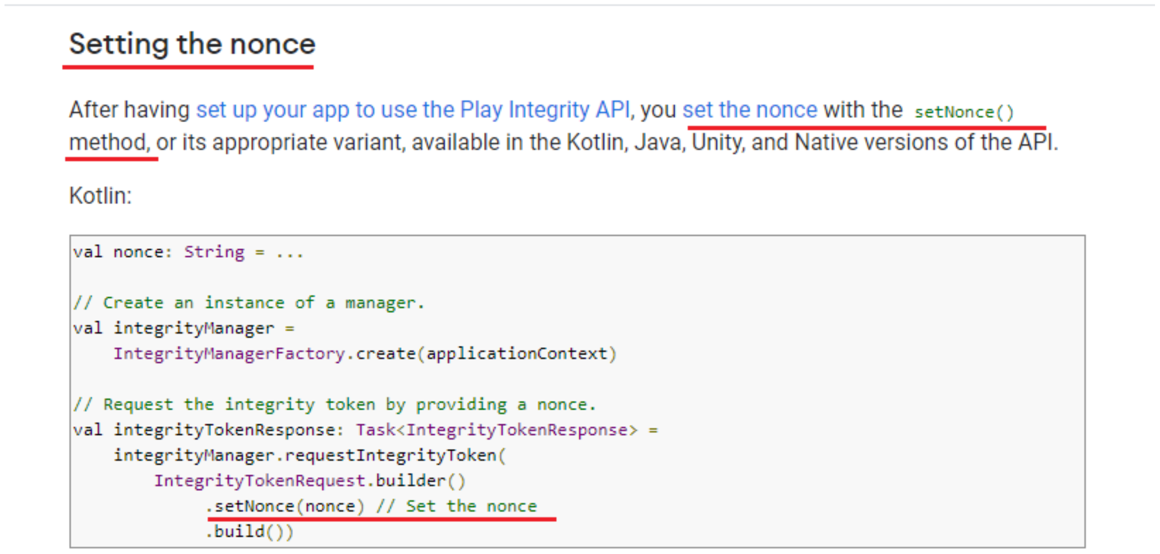

User action or server request that you want to check
Your app requests a Play Integrity API assessment
Play returns verdicts about device, app, and account
Your backend server decides what to do next

	https://developer.android.com/google/play/integrity/overview  If you choose to manage and download your response encryption keys, you can decrypt and verify the returned token within your own secure server environment. You can obtain the returned token by using the `IntegrityTokenResponse#token()` method. https://developer.android.com/google/play/integrity/classic#nonce  1. The user initiates the high-value action. 2. Your app asks the server for a unique value to identify the request 3. Your app server generates a globally unique value in a way that malicious users cannot predict. For example, you may use a cryptographically-secure random number generator to create such a value. We recommend creating values 128 bits or larger. 4. Your app server sends the globally unique value to the app. 5. Your app prepares a message it wants to protect, for example, in JSON format. 6. Your app calculates a cryptographic hash of the message it wants to protect. For example, with the SHA-256, or the SHA-3-256 hashing algorithms. 7. Your app creates a string by appending the unique value received from your app server, and the hash of the message it wants to protect. 8. Your app calls the Play Integrity API, and calls setNonce() to set the nonce field to the string created in the previous step. 9. Your app sends both the message it wants to protect, and the signed result of the Play Integrity API to your server. 10. Your app server splits the value of the nonce field, and verifies that the cryptographic hash of the message, as well as the unique value it previously generated match to the expected values, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

	 Your app server Your app Google Play Generate a globally-unique value Integrated Software Computer Program Send the unique value Software Application logic module Call `setNonce()` with the received value Call the Play Integrity API API returns signed result Send the signed result Operational assurance logic module Verify that the `nonce` field matches the unique value Your app server Your app Google Play https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1

	 1. The server creates a globally unique value in a way that malicious users cannot predict. For example, a cryptographically-secure random number 128 bits or larger. 2. Your app calls the Play Integrity API, and sets the nonce field to the unique value received by your app server. 3. Your app sends the signed result of the Play Integrity API to your server. 4. Your server verifies that the nonce field in the signed result matches the unique value it previously generated, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1 • A token generated by the Play Integrity API https://developer.android.com/codelabs/add-play-integrity#2

	 **What is a nonce?** In cryptography and security engineering, a nonce (*number once*) is a number that is used only once in a secure communication. There are many applications for nonces, such as in authentication, encryption and hashing. In the Play Integrity API, the nonce is an opaque base-64 encoded binary blob that you set before invoking the API integrity check, and it will be returned as-is inside the signed response of the API. Depending on how you create and validate the nonce, it is possible to leverage it to further strengthen the existing protections the Play Integrity API offers, as well as mitigate certain types of attacks, such as person-in-the-middle (PITM) tampering attacks, and replay attacks. Apart from returning the nonce as-is in the signed response, the Play Integrity API doesn't perform any processing of the actual nonce data, so as long as it is a valid base-64 value, you can set any arbitrary value. That said, in order to digitally sign the response, the nonce is sent to Google's servers, so it is very important not to set the nonce to any type of personally identifiable information (PII), such as the user's name, phone or email address. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html  **Nonce generation** Making a Play Integrity request requires generating a nonce and associating it with the request. The nonce is used to help ensure an integrity request is unique and only processed once. The nonce is also used to verify the message contents associated with the integrity request haven't been tampered with. For more information about Play Integrity nonces, see the documentation. In this codelab, the nonce is generated by combining two values: • A 128-bit random number generated by a cryptographically-secure random number generator • A SHA-256 hash of the `commandString` value

	https://developer.android.com/codelabs/add-play-integrity#2 

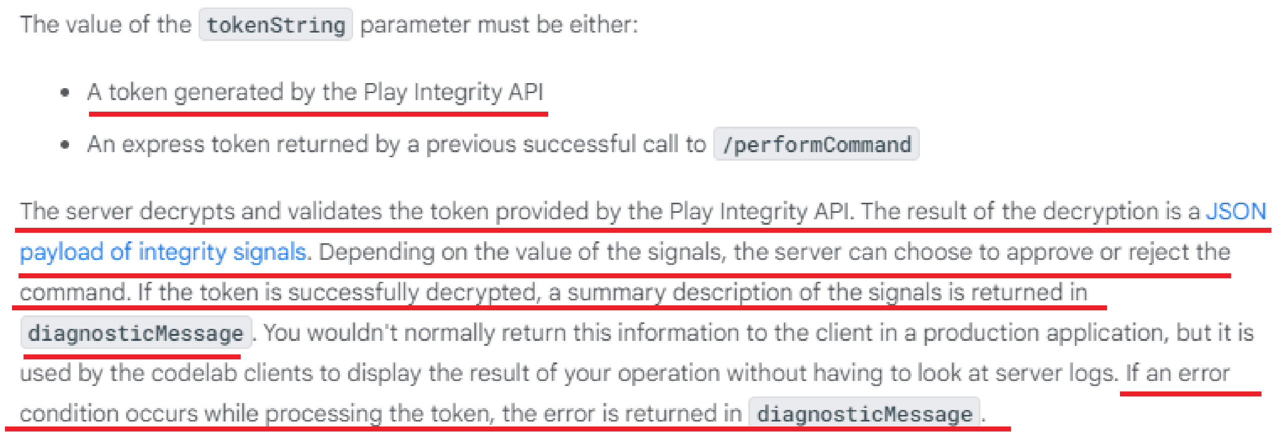
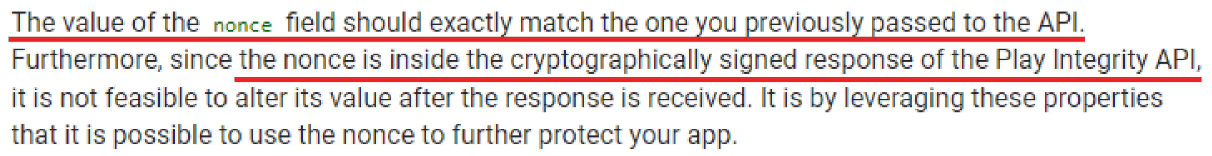
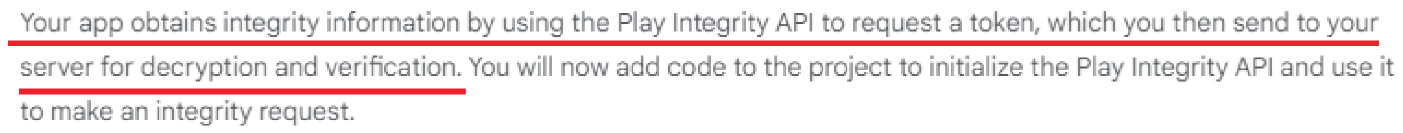
Setting the nonce

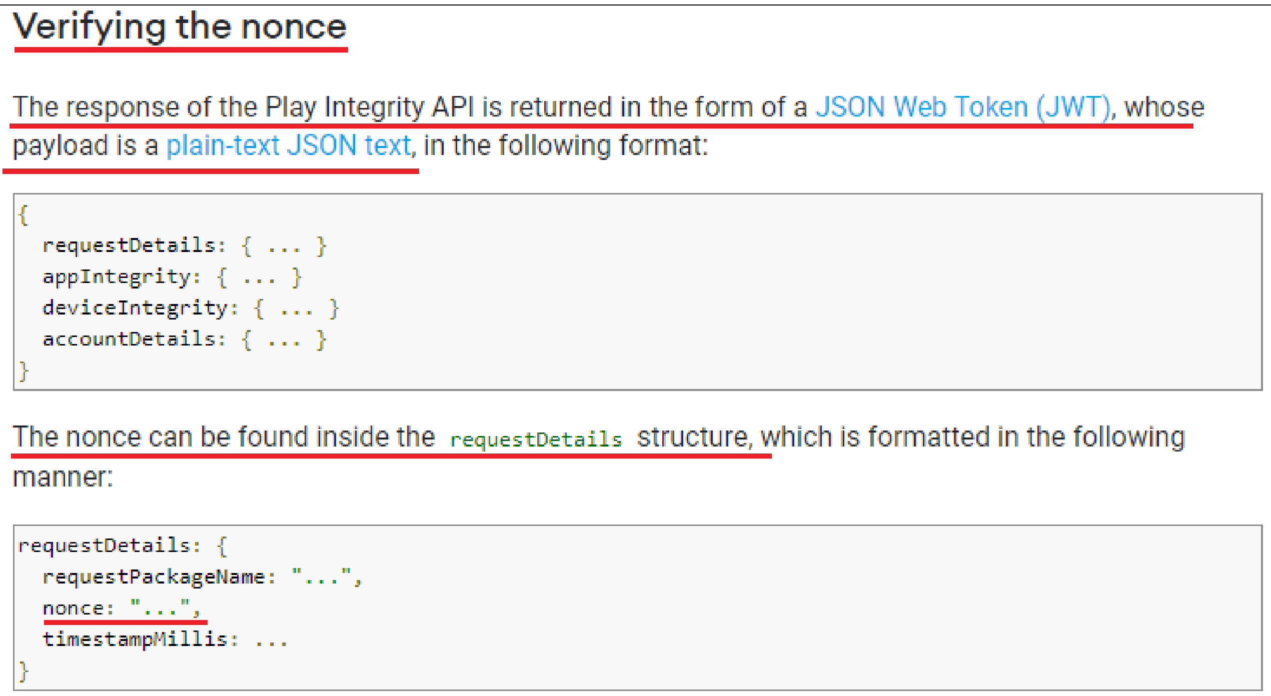
After having set up your app to use the Play Integrity API, you set the nonce with the `setNonce()` method, or its appropriate variant, available in the Kotlin, Java, Unity, and Native versions of the API.

Kotlin:

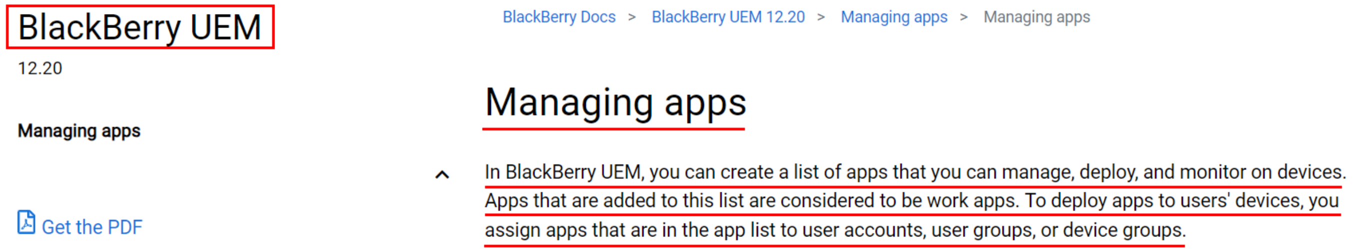
```
val nonce: String = ...

// Create an instance of a manager.
val integrityManager =
    IntegrityManagerFactory.create(applicationContext)

// Request the integrity token by providing a nonce.
val integrityTokenResponse: Task<IntegrityTokenResponse> =
    integrityManager.requestIntegrityToken(
        IntegrityTokenRequest.builder()
             .setNonce(nonce) // Set the nonce
             .build())
```

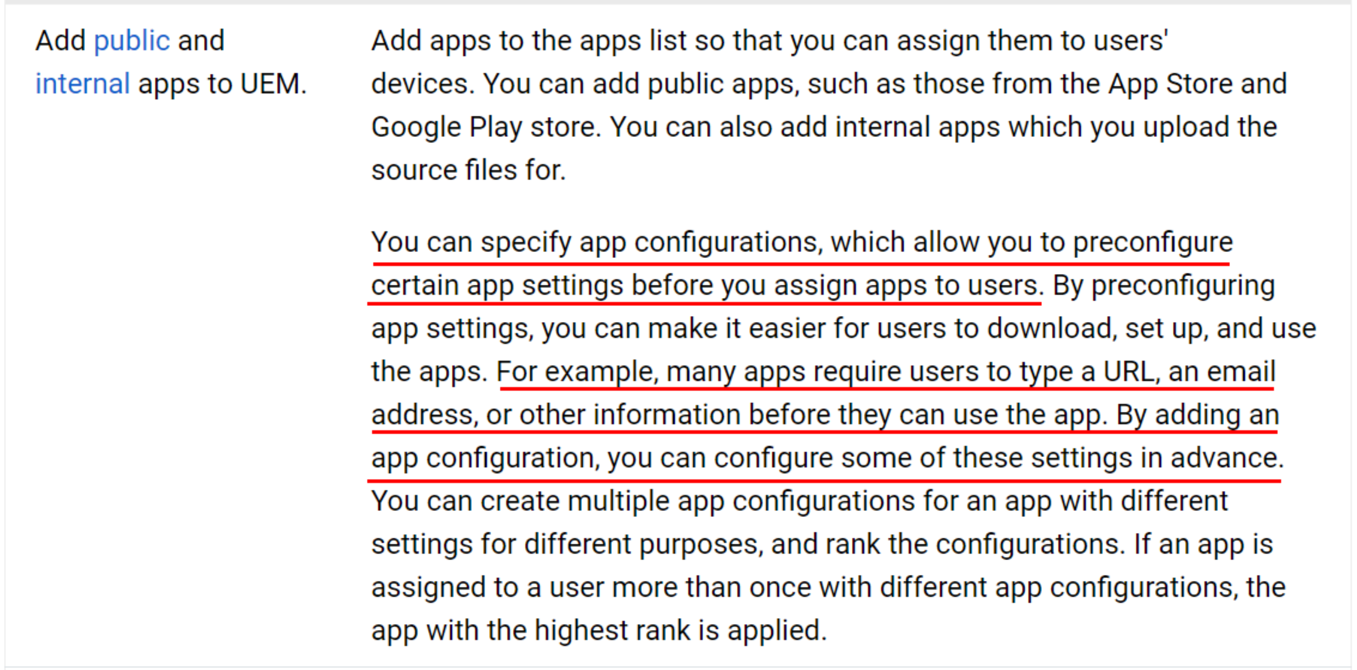
https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

	 The value of the `tokenString` parameter must be either: • A token generated by the Play Integrity API • An express token returned by a previous successful call to `/performCommand` The server decrypts and validates the token provided by the Play Integrity API. The result of the decryption is a JSON payload of integrity signals. Depending on the value of the signals, the server can choose to approve or reject the command. If the token is successfully decrypted, a summary description of the signals is returned in `diagnosticMessage`. You wouldn't normally return this information to the client in a production application, but it is used by the codelab clients to display the result of your operation without having to look at server logs. If an error condition occurs while processing the token, the error is returned in `diagnosticMessage`. https://developer.android.com/codelabs/add-play-integrity#2  The value of the `nonce` field should exactly match the one you previously passed to the API. Furthermore, since the nonce is inside the cryptographically signed response of the Play Integrity API, it is not feasible to alter its value after the response is received. It is by leveraging these properties that it is possible to use the nonce to further protect your app. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html  Your app obtains integrity information by using the Play Integrity API to request a token, which you then send to your server for decryption and verification. You will now add code to the project to initialize the Play Integrity API and use it to make an integrity request. https://developer.android.com/codelabs/add-play-integrity#8

	 **Verifying the nonce**
	
	The response of the Play Integrity API is returned in the form of a JSON Web Token (JWT), whose payload is a plain-text JSON text, in the following format:
	
	`{`
	`  requestDetails: { ... }`
	`  appIntegrity: { ... }`
	`  deviceIntegrity: { ... }`
	`  accountDetails: { ... }`
	`}`
	
	The nonce can be found inside the `requestDetails` structure, which is formatted in the following manner:
	
	`requestDetails: {`
	`  requestPackageName: "...",`
	`  nonce: "...",`
	`  timestampMillis: ...`
	`}`
	
	https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

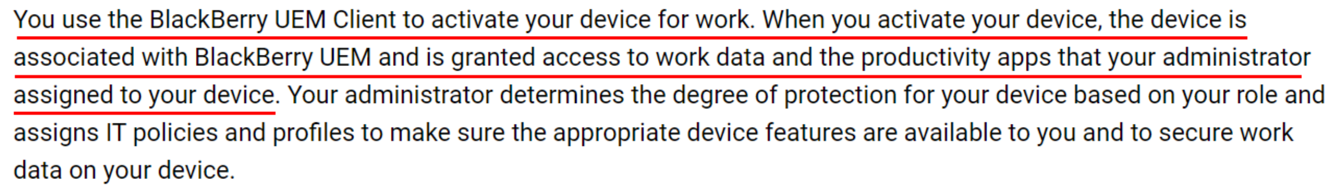
wherein the executing of the integrated software computer program provides a set of combined computing functions: wherein during the combined computing functions, an operational	The accused instrumentality discloses wherein the executing of the integrated software computer program (e.g., a work application along with attestation API, etc.) provides a set of combined computing functions (e.g., program instructions, app functions, etc.): wherein during the combined computing functions (e.g., program instructions, app functions, etc.), an operational set of unique data tags (e.g., a nonce, signed token, etc.) are generated by the sub-procedures (e.g., algorithms for Play Integrity Check API) of the operational assurance logic module (e.g., Attestation API, etc.) to provide for validating that the integrated software computer program (e.g., a work application along with attestation API, etc.) as executed was not tampered with.

set of unique data tags are generated by the sub-procedures of the operational assurance logic module to provide for validating that the integrated software computer program as executed was not tampered with; **Col 28: lines 14-20** *The resulting system is such wherein the intermixture is further comprised of at least one of: obfuscation, encryption, replication, adding dummy code, addition of redundant control, renaming of variables,* ***<u>splitting a procedure into multiple sub-procedure,</u>*** *dictionary transformation, compilation, interpretation, cryptographic transformation, digital signing, and scrambling*	As shown below, the accused instrumentality is a Unified Endpoint Management (UEM) solution that allows enterprises to manage and control devices linked to employees. It allows administrators to add work applications that are permitted for use by employees on their managed devices. Also, the accused instrumentality utilizes device attestation for verifying device & app integrity. When execution of a work application (e.g., a software computer program) running on an employee device requires integrity verification, it utilizes device attestation feature for the verification. It utilizes an attestation API (e.g., an operation assurance logic module) such as Play Integrity API for an android device of an employee. The work app sends a request to the attestation API, it sends a signed token (e.g., unique security tag, etc.) back to the app. The app forwards the signed token to a server for verification. The server verifies whether the signed token matches its previously generated unique value. Thus, if the values match, it determines the valid app is executing a program on a valid device. BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps Managing apps In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps



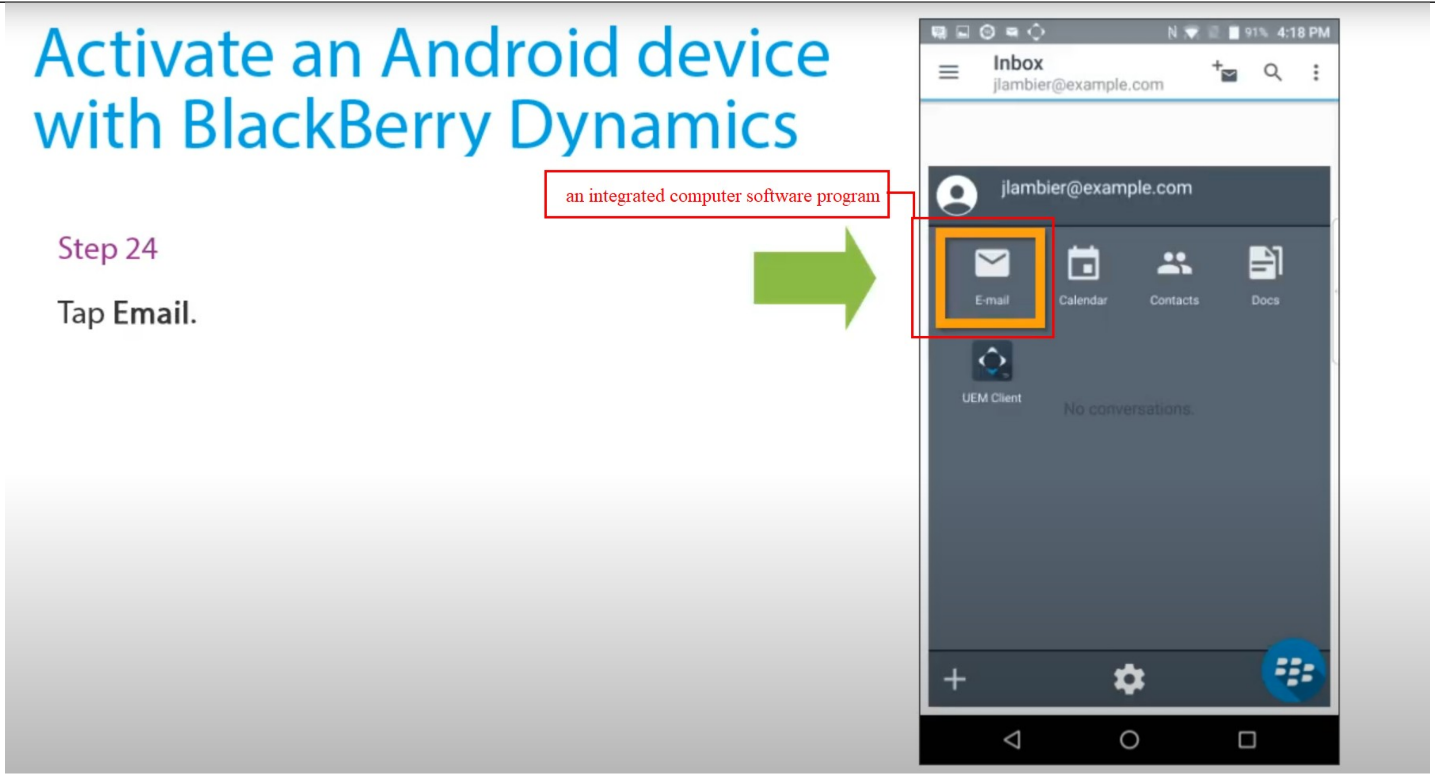
Add public and internal apps to UEM.	Add apps to the apps list so that you can assign them to users' devices. You can add public apps, such as those from the App Store and Google Play store. You can also add internal apps which you upload the source files for. You can specify app configurations, which allow you to preconfigure certain app settings before you assign apps to users. By preconfiguring app settings, you can make it easier for users to download, set up, and use the apps. For example, many apps require users to type a URL, an email address, or other information before they can use the app. By adding an app configuration, you can configure some of these settings in advance. You can create multiple app configurations for an app with different settings for different purposes, and rank the configurations. If an app is assigned to a user more than once with different app configurations, the app with the highest rank is applied.

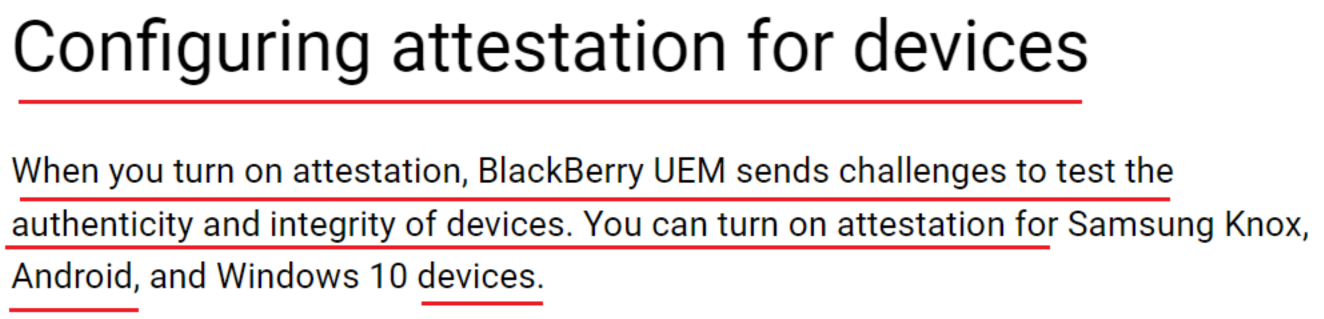
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

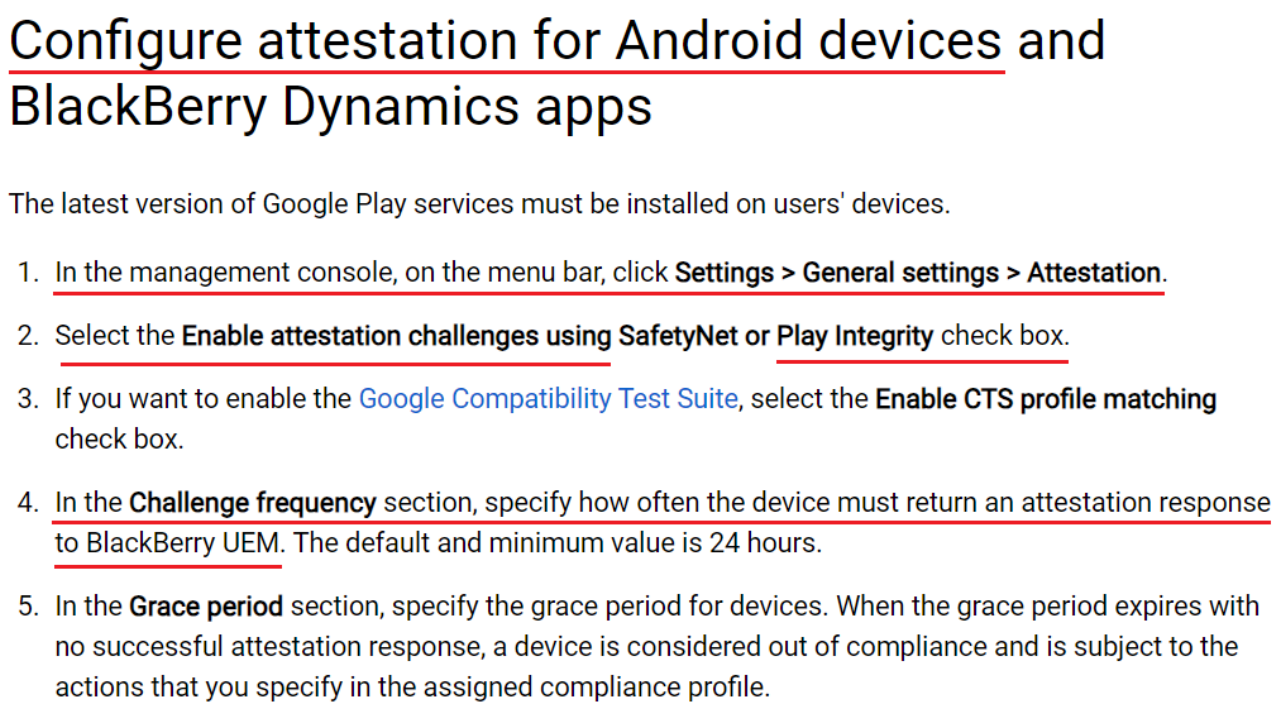


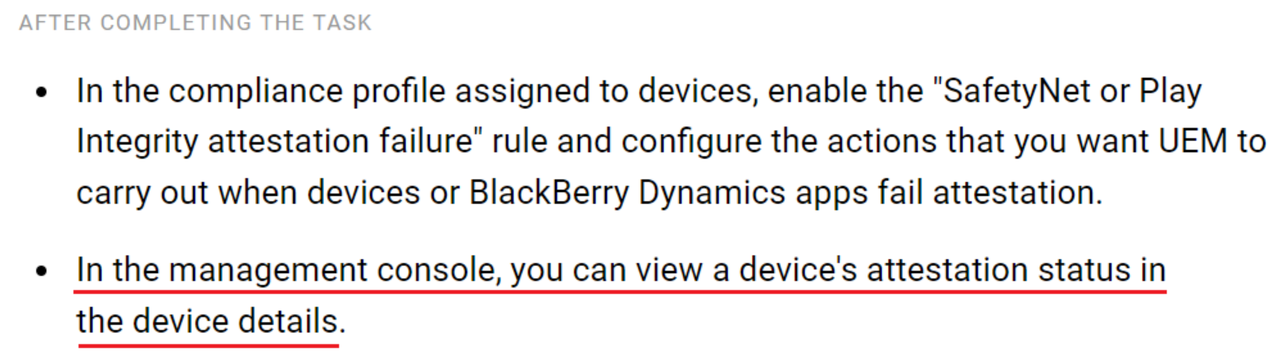
You use the BlackBerry UEM Client to activate your device for work. When you activate your device, the device is associated with BlackBerry UEM and is granted access to work data and the productivity apps that your administrator assigned to your device. Your administrator determines the degree of protection for your device based on your role and assigns IT policies and profiles to make sure the appropriate device features are available to you and to secure work data on your device.

https://docs.blackberry.com/en/endpoint-management/blackberry-uem-client/latest#:~:text=You%20use%20the%20BlackBerry%20UEM,work%20data%20on%20your%20device.

	 Activate an Android device with BlackBerry Dynamics Step 24 Tap **Email.** an integrated computer software program Inbox jlambier@example.com jlambier@example.com E-mail Calendar Contacts Docs UEM Client No conversations. https://www.youtube.com/watch?v=zRrI1TpLVXM

	 Configuring attestation for devices When you turn on attestation, BlackBerry UEM sends challenges to test the authenticity and integrity of devices. You can turn on attestation for Samsung Knox, Android, and Windows 10 devices. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation

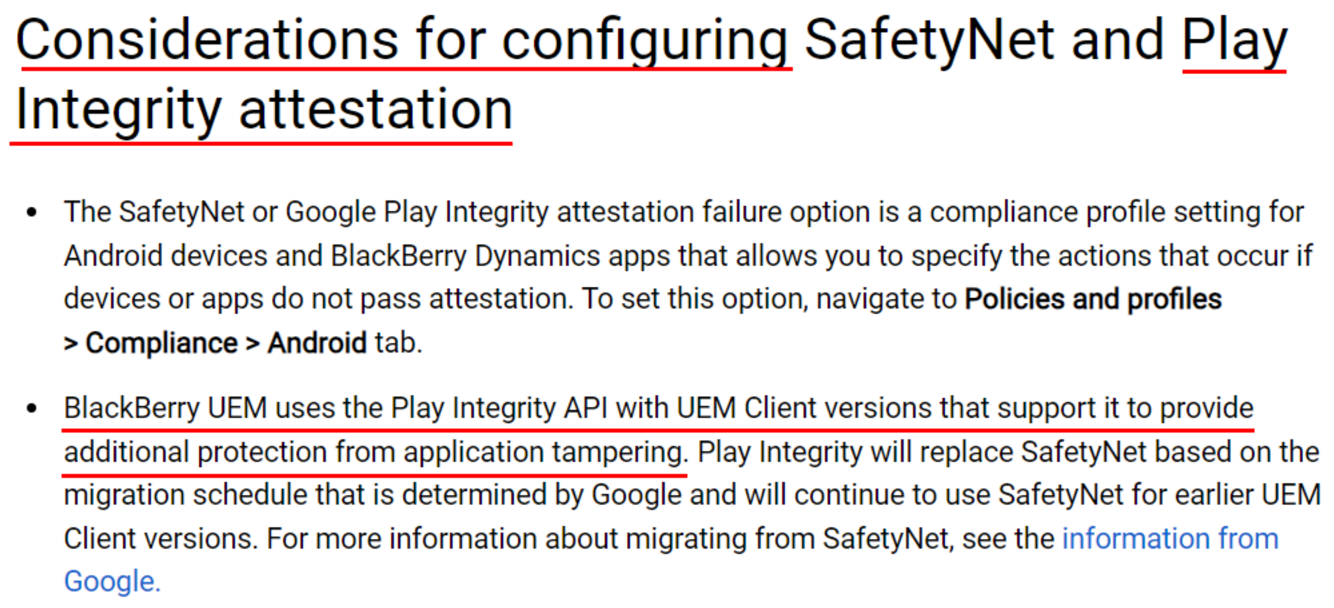
	 **Configure attestation for Android devices and BlackBerry Dynamics apps** The latest version of Google Play services must be installed on users' devices. 1. In the management console, on the menu bar, click **Settings > General settings > Attestation**. 2. Select the **Enable attestation challenges using SafetyNet or Play Integrity** check box. 3. If you want to enable the Google Compatibility Test Suite, select the **Enable CTS profile matching** check box. 4. In the **Challenge frequency** section, specify how often the device must return an attestation response to BlackBerry UEM. The default and minimum value is 24 hours. 5. In the **Grace period** section, specify the grace period for devices. When the grace period expires with no successful attestation response, a device is considered out of compliance and is subject to the actions that you specify in the assigned compliance profile. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841



AFTER COMPLETING THE TASK

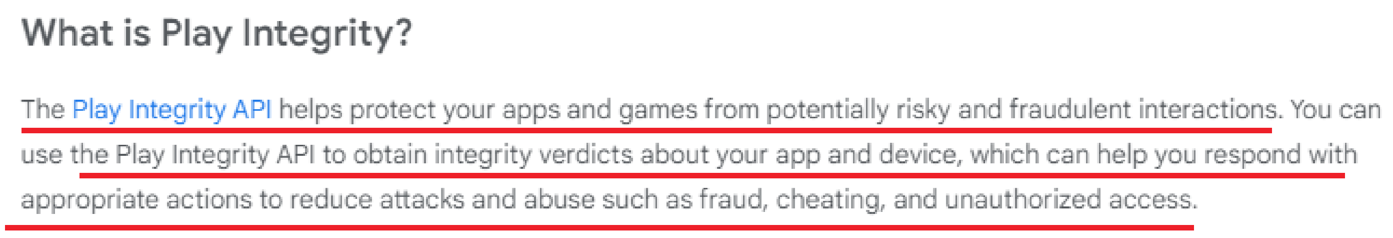
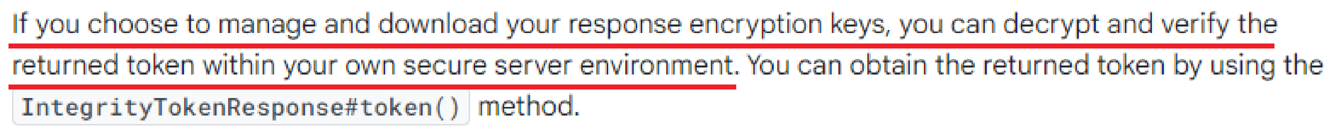
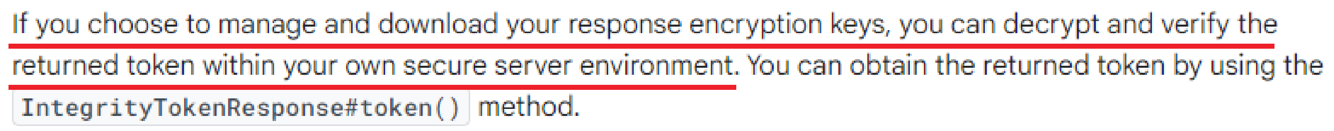
- In the compliance profile assigned to devices, enable the "SafetyNet or Play Integrity attestation failure" rule and configure the actions that you want UEM to carry out when devices or BlackBerry Dynamics apps fail attestation.
- In the management console, you can view a device's attestation status in the device details.

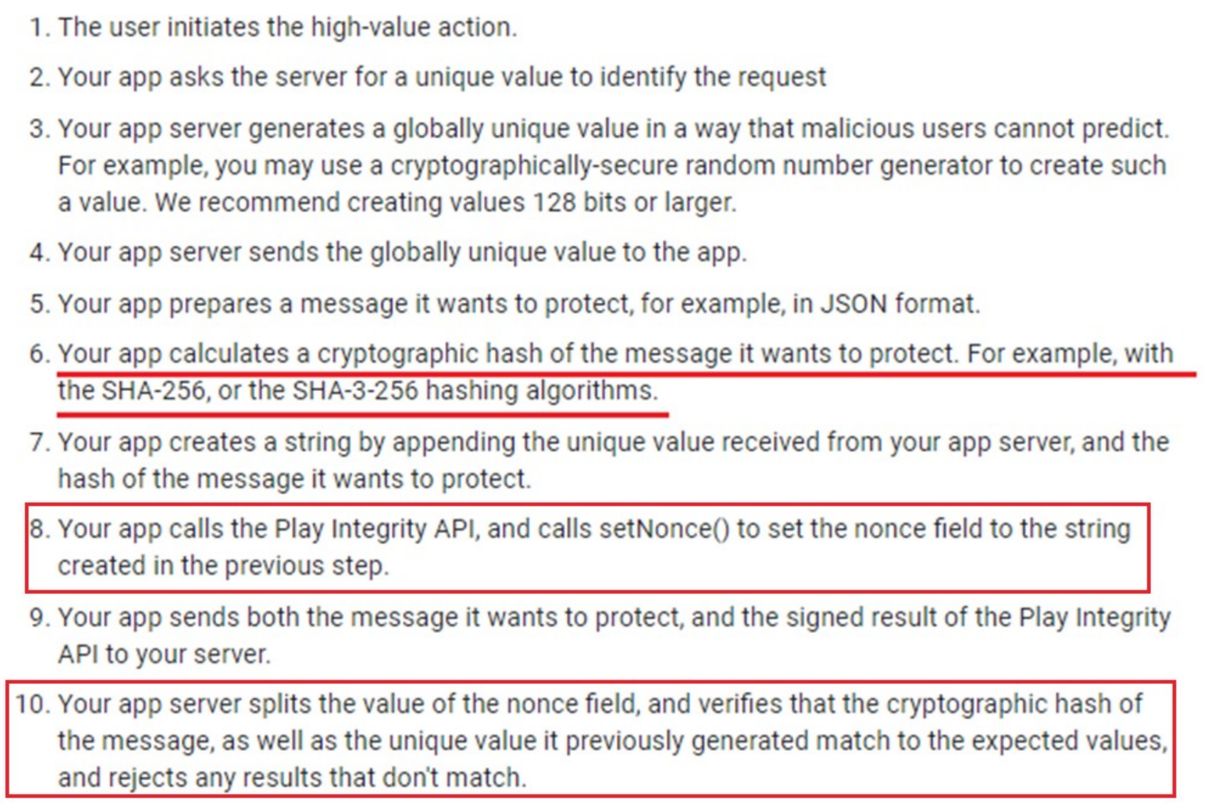
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841

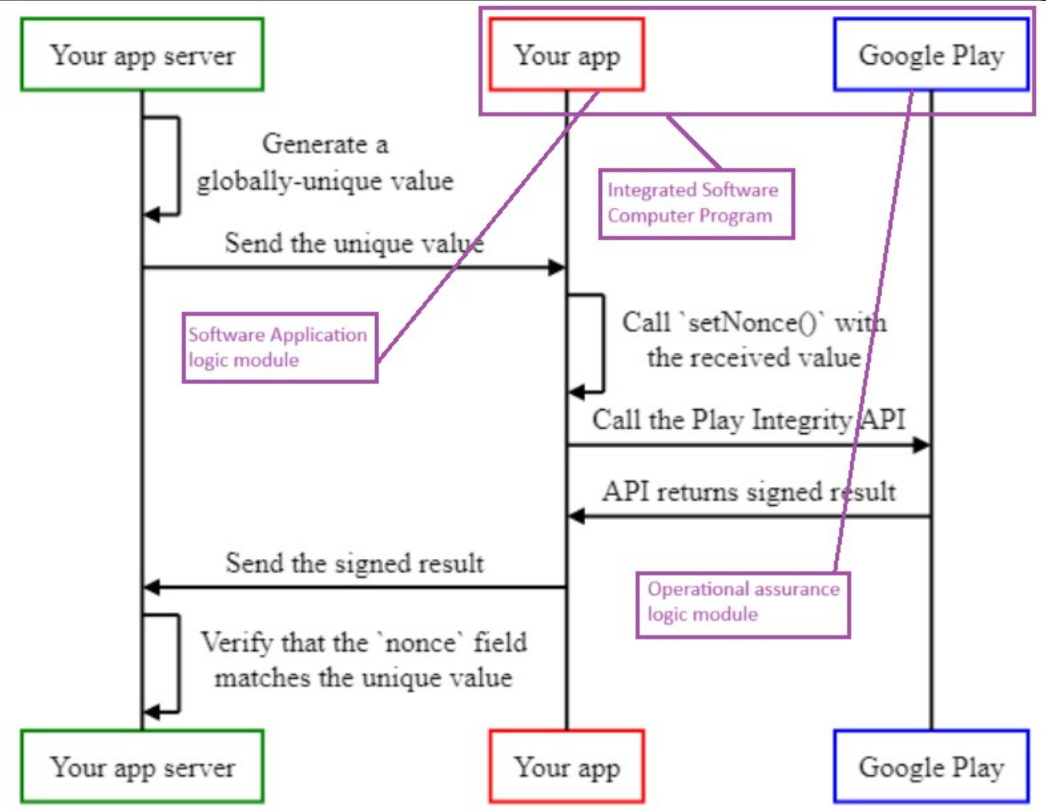


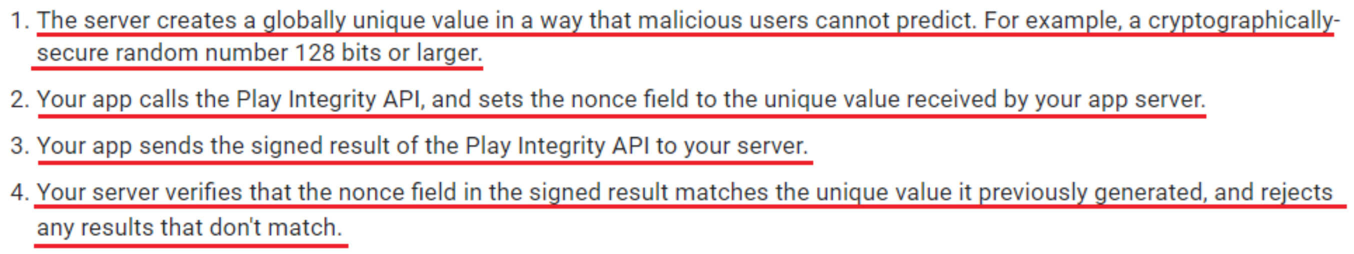
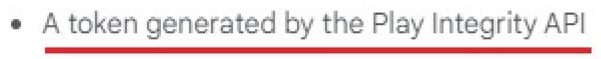
Considerations for configuring SafetyNet and Play Integrity attestation

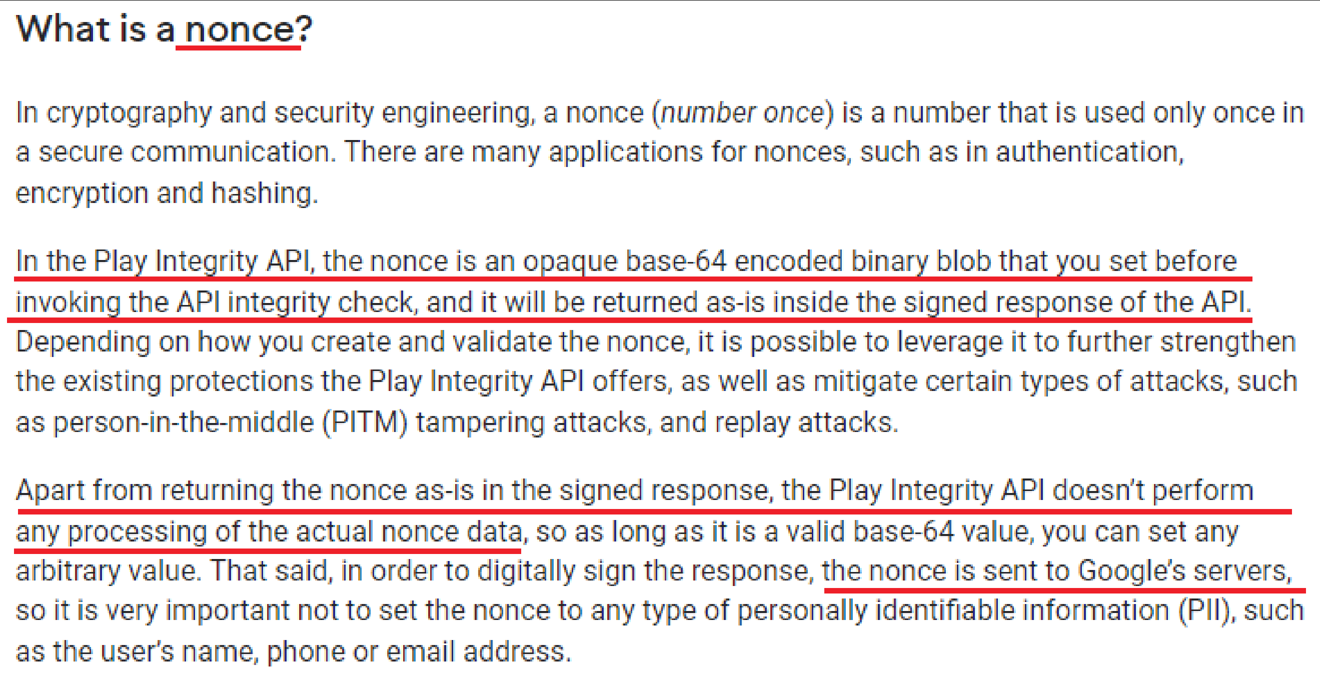
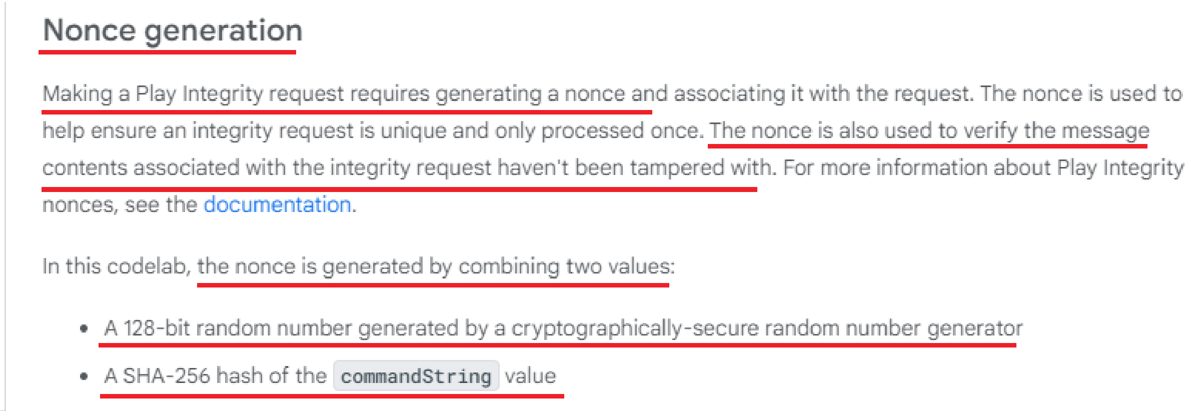
- The SafetyNet or Google Play Integrity attestation failure option is a compliance profile setting for Android devices and BlackBerry Dynamics apps that allows you to specify the actions that occur if devices or apps do not pass attestation. To set this option, navigate to **Policies and profiles > Compliance > Android** tab.
- BlackBerry UEM uses the Play Integrity API with UEM Client versions that support it to provide additional protection from application tampering. Play Integrity will replace SafetyNet based on the migration schedule that is determined by Google and will continue to use SafetyNet for earlier UEM Client versions. For more information about migrating from SafetyNet, see the information from Google.

	https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_18/administration/device-features-it-policies/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/Considerations-for-configuring-attestation-for-Android-devices-and-apps-using-SafetyNet  **What is Play Integrity?** The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions. You can use the Play Integrity API to obtain integrity verdicts about your app and device, which can help you respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access. https://developer.android.com/codelabs/add-play-integrity#0  If you choose to manage and download your response encryption keys, you can decrypt and verify the returned token within your own secure server environment. You can obtain the returned token by using the `IntegrityTokenResponse#token()` method. https://developer.android.com/google/play/integrity/classic#nonce  If you choose to manage and download your response encryption keys, you can decrypt and verify the returned token within your own secure server environment. You can obtain the returned token by using the `IntegrityTokenResponse#token()` method. https://developer.android.com/google/play/integrity/classic#nonce

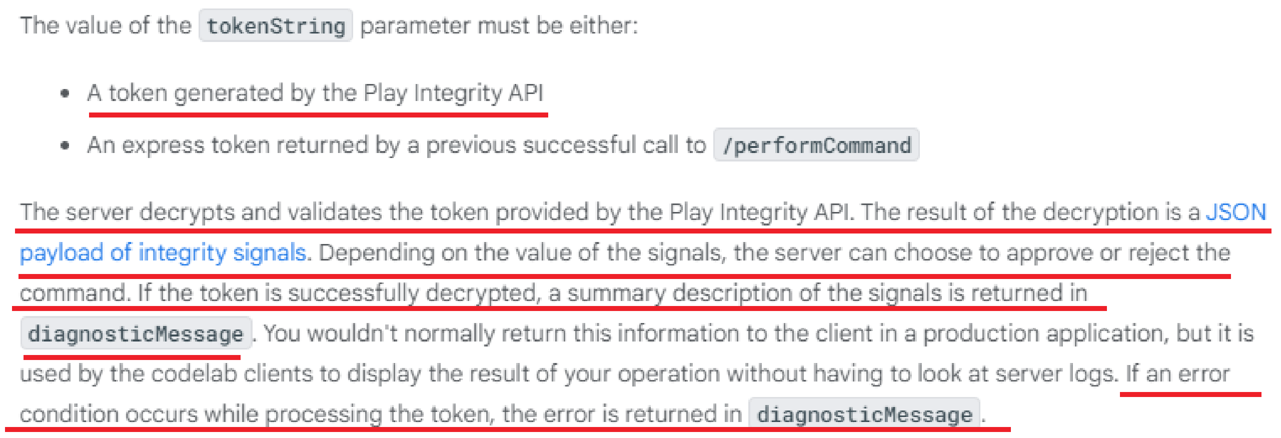
	 1. The user initiates the high-value action. 2. Your app asks the server for a unique value to identify the request 3. Your app server generates a globally unique value in a way that malicious users cannot predict. For example, you may use a cryptographically-secure random number generator to create such a value. We recommend creating values 128 bits or larger. 4. Your app server sends the globally unique value to the app. 5. Your app prepares a message it wants to protect, for example, in JSON format. 6. Your app calculates a cryptographic hash of the message it wants to protect. For example, with the SHA-256, or the SHA-3-256 hashing algorithms. 7. Your app creates a string by appending the unique value received from your app server, and the hash of the message it wants to protect. 8. Your app calls the Play Integrity API, and calls setNonce() to set the nonce field to the string created in the previous step. 9. Your app sends both the message it wants to protect, and the signed result of the Play Integrity API to your server. 10. Your app server splits the value of the nonce field, and verifies that the cryptographic hash of the message, as well as the unique value it previously generated match to the expected values, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

	 Your app server Your app Google Play Generate a globally-unique value Integrated Software Computer Program Send the unique value Call `setNonce()` with the received value Software Application logic module Call the Play Integrity API API returns signed result Send the signed result Operational assurance logic module Verify that the `nonce` field matches the unique value Your app server Your app Google Play https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1

	 1. The server creates a globally unique value in a way that malicious users cannot predict. For example, a cryptographically-secure random number 128 bits or larger. 2. Your app calls the Play Integrity API, and sets the nonce field to the unique value received by your app server. 3. Your app sends the signed result of the Play Integrity API to your server. 4. Your server verifies that the nonce field in the signed result matches the unique value it previously generated, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1  • A token generated by the Play Integrity API https://developer.android.com/codelabs/add-play-integrity#2

	 **What is a nonce?** In cryptography and security engineering, a nonce (*number once*) is a number that is used only once in a secure communication. There are many applications for nonces, such as in authentication, encryption and hashing. In the Play Integrity API, the nonce is an opaque base-64 encoded binary blob that you set before invoking the API integrity check, and it will be returned as-is inside the signed response of the API. Depending on how you create and validate the nonce, it is possible to leverage it to further strengthen the existing protections the Play Integrity API offers, as well as mitigate certain types of attacks, such as person-in-the-middle (PITM) tampering attacks, and replay attacks. Apart from returning the nonce as-is in the signed response, the Play Integrity API doesn't perform any processing of the actual nonce data, so as long as it is a valid base-64 value, you can set any arbitrary value. That said, in order to digitally sign the response, the nonce is sent to Google's servers, so it is very important not to set the nonce to any type of personally identifiable information (PII), such as the user's name, phone or email address. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html  **Nonce generation** Making a Play Integrity request requires generating a nonce and associating it with the request. The nonce is used to help ensure an integrity request is unique and only processed once. The nonce is also used to verify the message contents associated with the integrity request haven't been tampered with. For more information about Play Integrity nonces, see the documentation. In this codelab, the nonce is generated by combining two values: • A 128-bit random number generated by a cryptographically-secure random number generator • A SHA-256 hash of the `commandString` value

https://developer.android.com/codelabs/add-play-integrity#2

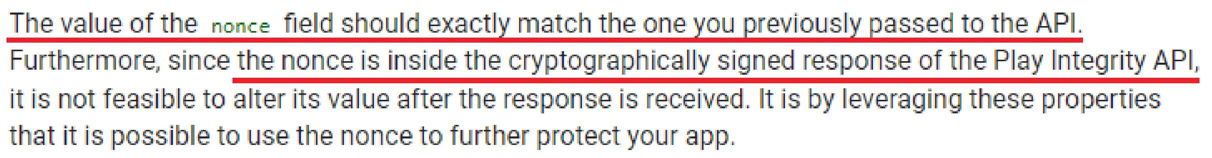


The value of the `tokenString` parameter must be either:

- A token generated by the Play Integrity API
- An express token returned by a previous successful call to `/performCommand`

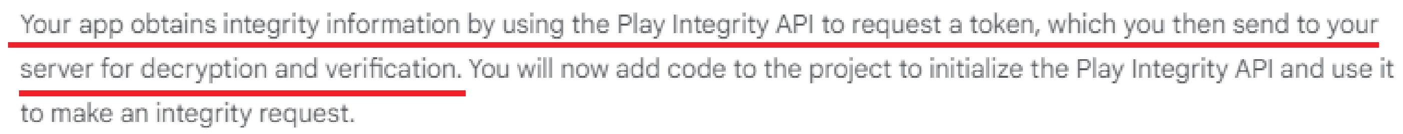
The server decrypts and validates the token provided by the Play Integrity API. The result of the decryption is a JSON payload of integrity signals. Depending on the value of the signals, the server can choose to approve or reject the command. If the token is successfully decrypted, a summary description of the signals is returned in `diagnosticMessage`. You wouldn't normally return this information to the client in a production application, but it is used by the codelab clients to display the result of your operation without having to look at server logs. If an error condition occurs while processing the token, the error is returned in `diagnosticMessage`.

https://developer.android.com/codelabs/add-play-integrity#2



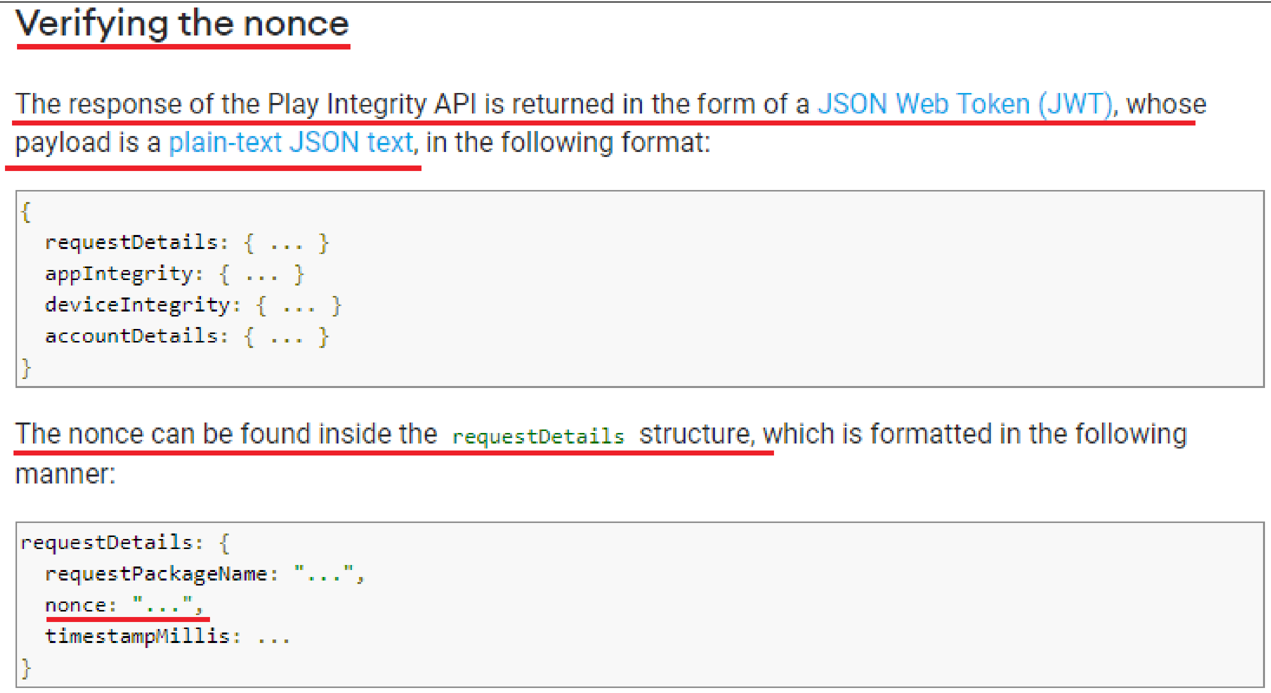
The value of the `nonce` field should exactly match the one you previously passed to the API. Furthermore, since the nonce is inside the cryptographically signed response of the Play Integrity API, it is not feasible to alter its value after the response is received. It is by leveraging these properties that it is possible to use the nonce to further protect your app.

https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html



Your app obtains integrity information by using the Play Integrity API to request a token, which you then send to your server for decryption and verification. You will now add code to the project to initialize the Play Integrity API and use it to make an integrity request.

https://developer.android.com/codelabs/add-play-integrity#8

<table>
<tr><td></td><td>

Verifying the nonce

The response of the Play Integrity API is returned in the form of a JSON Web Token (JWT), whose payload is a plain-text JSON text, in the following format:

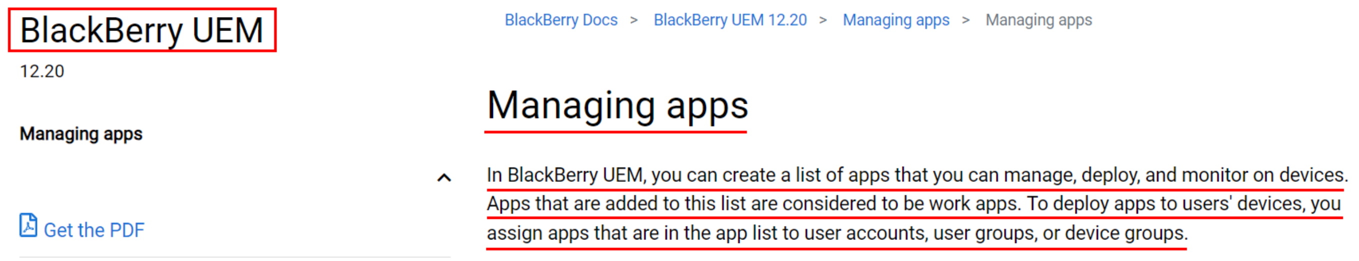
```
{
  requestDetails: { ... }
  appIntegrity: { ... }
  deviceIntegrity: { ... }
  accountDetails: { ... }
}
```

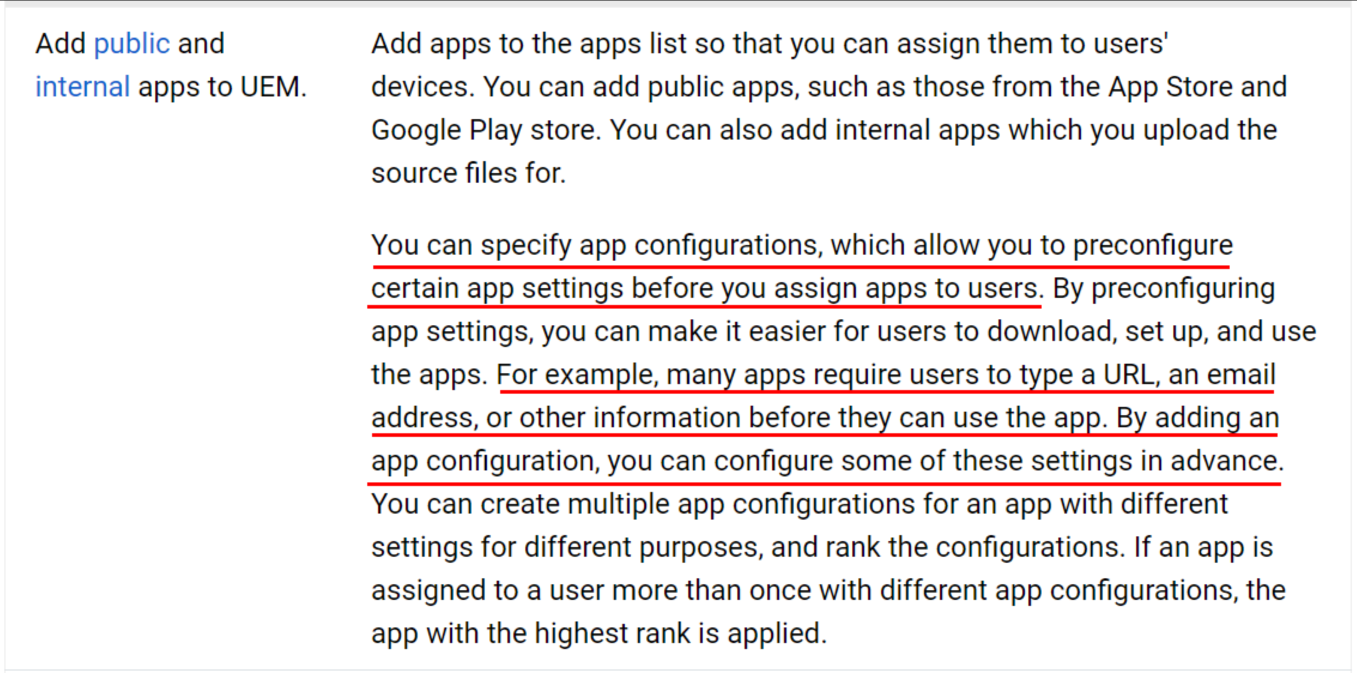
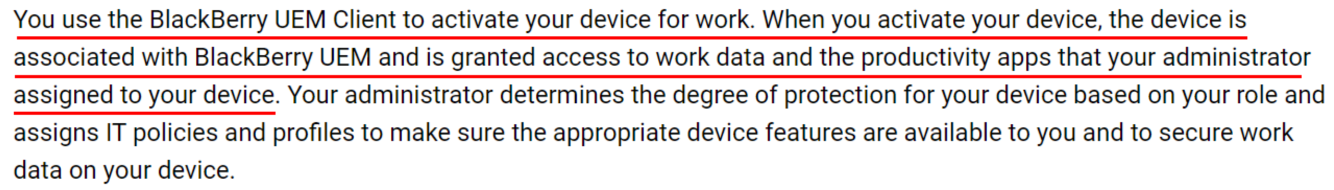
The nonce can be found inside the `requestDetails` structure, which is formatted in the following manner:

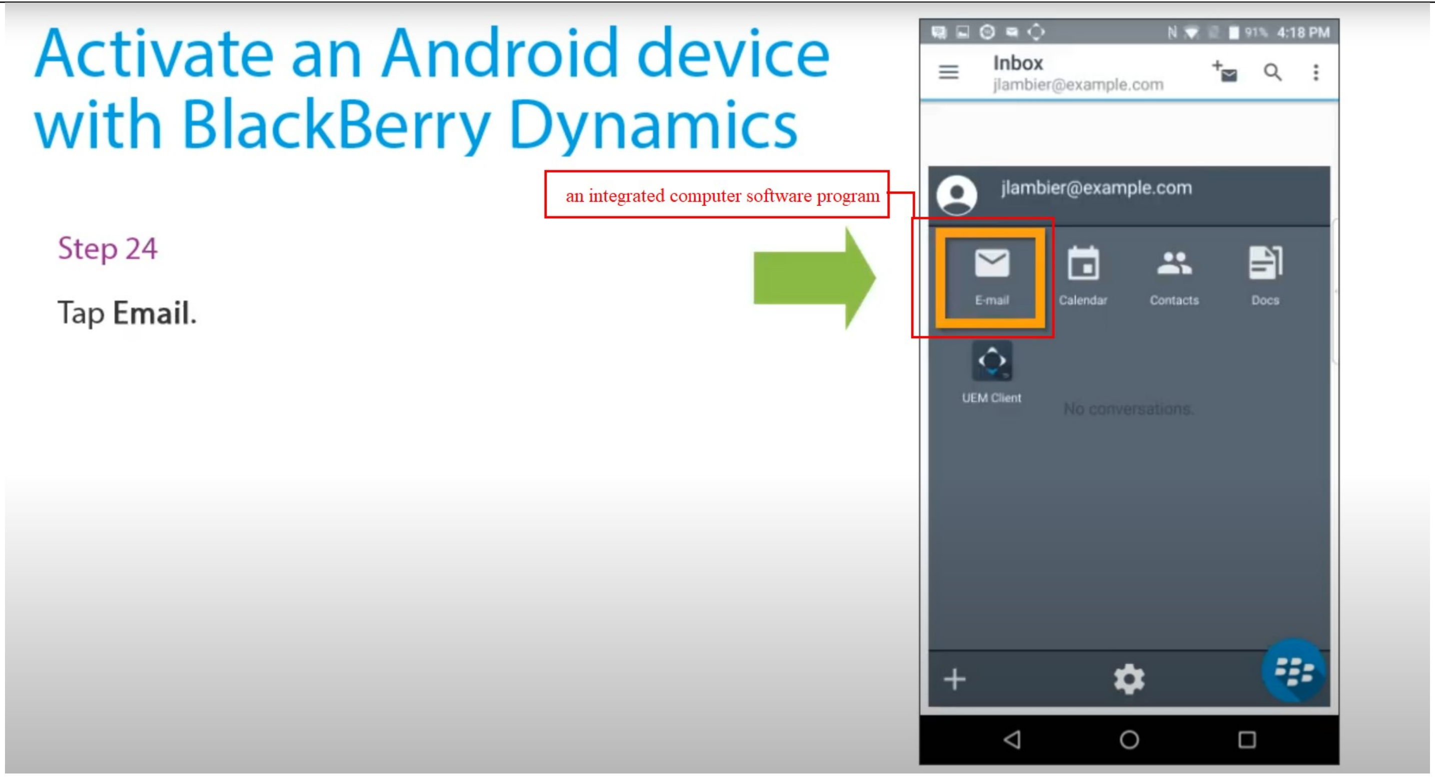
```
requestDetails: {
  requestPackageName: "...",
  nonce: "...",
  timestampMillis: ...
}
```

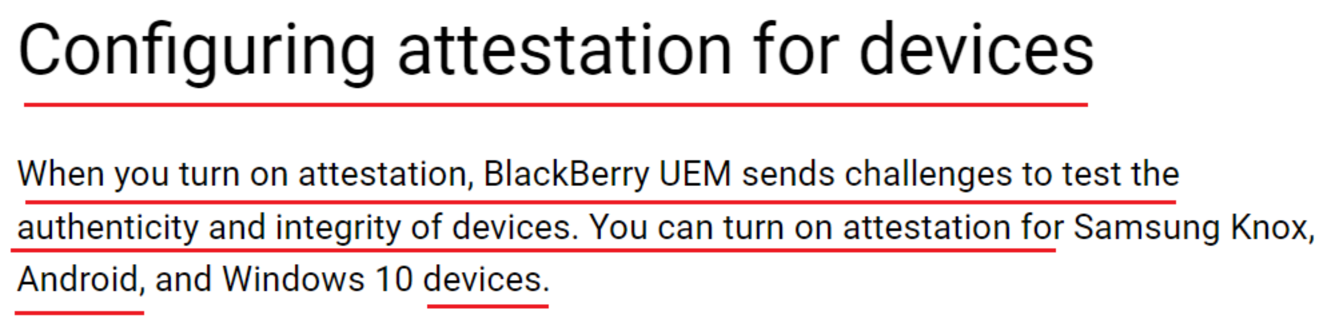
https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html</td></tr>
<tr><td>wherein the integrated software computer program generates the operational unique data tags which only when operating as a part of the integrated software computer program; and</td><td>The accused instrumentality discloses wherein the integrated software computer program (e.g., a work application along with attestation API, etc.) generates the operational unique data tags (e.g., a nonce, signed token, etc.) which only when operating as a part of the integrated software computer program (e.g., a work application along with attestation API, etc.).

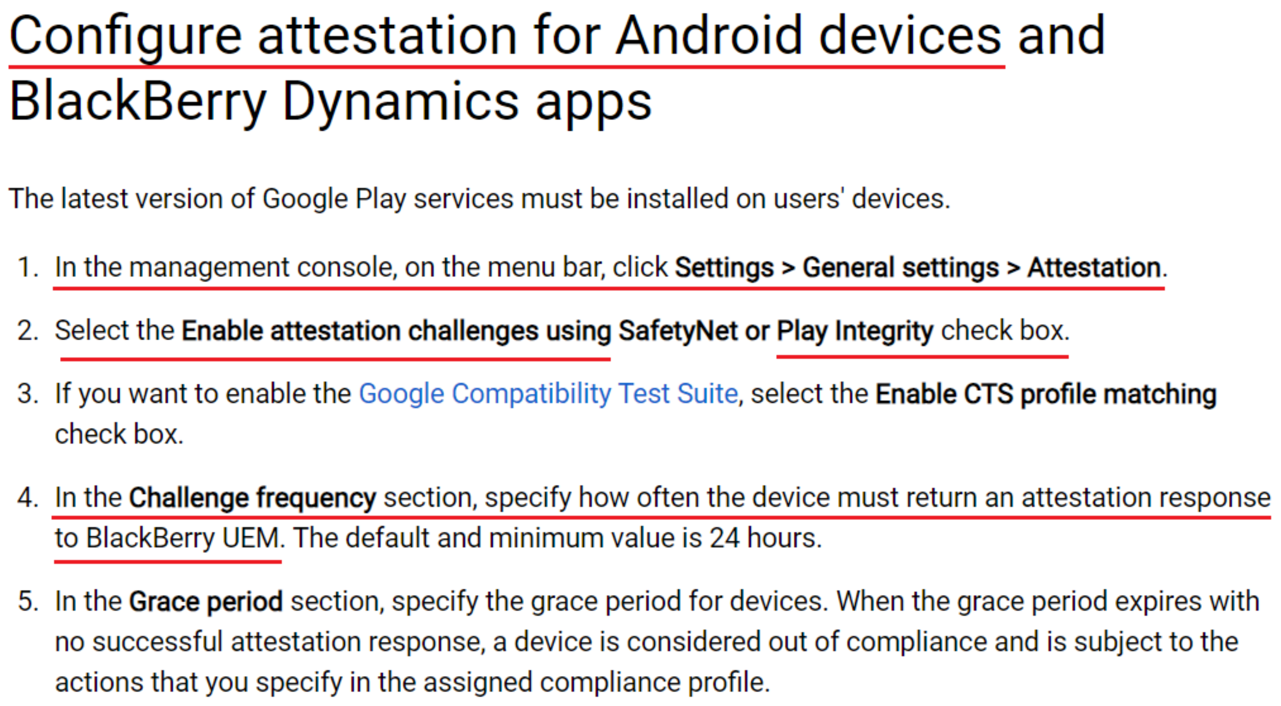
As shown below, the accused instrumentality is a Unified Endpoint Management (UEM) solution that allows enterprises to manage and control devices linked to employees. It allows administrators to add work applications that are permitted for use by employees on their managed devices.</td></tr>
</table>

	Also, the accused instrumentality utilizes device attestation for verifying device & app integrity. When execution of a work application (e.g., a software computer program) running on an employee device requires integrity verification, it utilizes device attestation feature for the verification. It utilizes an attestation API (e.g., an operation assurance logic module) such as Play Integrity API for an android device of an employee. The work app sends a request to the attestation API, it sends a signed token (e.g., unique security tag, etc.) back to the app. The app forwards the signed token to a server for verification. The server verifies whether the signed token matches its previously generated unique value. Thus, if the values match, it determines the valid app is executing a program on a valid device.  BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps Managing apps In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

	 Add public and internal apps to UEM. — Add apps to the apps list so that you can assign them to users' devices. You can add public apps, such as those from the App Store and Google Play store. You can also add internal apps which you upload the source files for. You can specify app configurations, which allow you to preconfigure certain app settings before you assign apps to users. By preconfiguring app settings, you can make it easier for users to download, set up, and use the apps. For example, many apps require users to type a URL, an email address, or other information before they can use the app. By adding an app configuration, you can configure some of these settings in advance. You can create multiple app configurations for an app with different settings for different purposes, and rank the configurations. If an app is assigned to a user more than once with different app configurations, the app with the highest rank is applied. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps  You use the BlackBerry UEM Client to activate your device for work. When you activate your device, the device is associated with BlackBerry UEM and is granted access to work data and the productivity apps that your administrator assigned to your device. Your administrator determines the degree of protection for your device based on your role and assigns IT policies and profiles to make sure the appropriate device features are available to you and to secure work data on your device. https://docs.blackberry.com/en/endpoint-management/blackberry-uem-client/latest#:~:text=You%20use%20the%20BlackBerry%20UEM,work%20data%20on%20your%20device.

	 Activate an Android device with BlackBerry Dynamics Step 24 Tap **Email.** an integrated computer software program Inbox jlambier@example.com jlambier@example.com E-mail Calendar Contacts Docs UEM Client No conversations. https://www.youtube.com/watch?v=zRrI1TpLVXM

	 Configuring attestation for devices When you turn on attestation, BlackBerry UEM sends challenges to test the authenticity and integrity of devices. You can turn on attestation for Samsung Knox, Android, and Windows 10 devices. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation

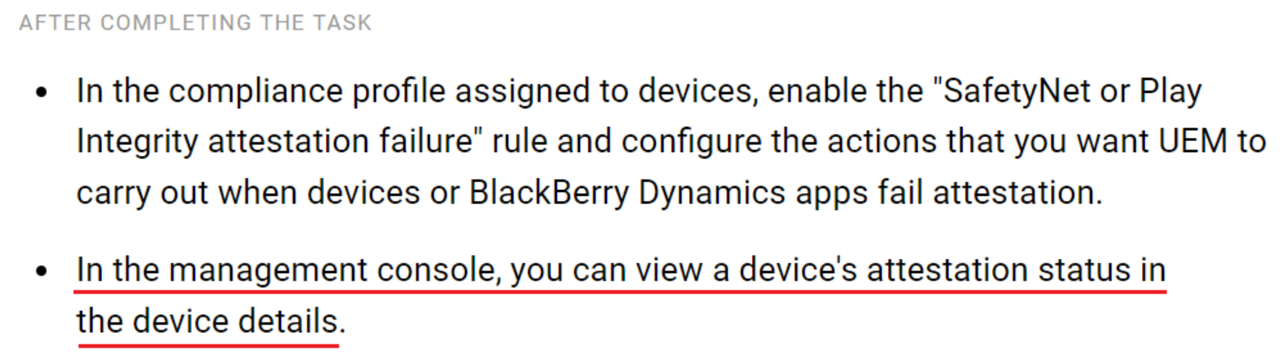
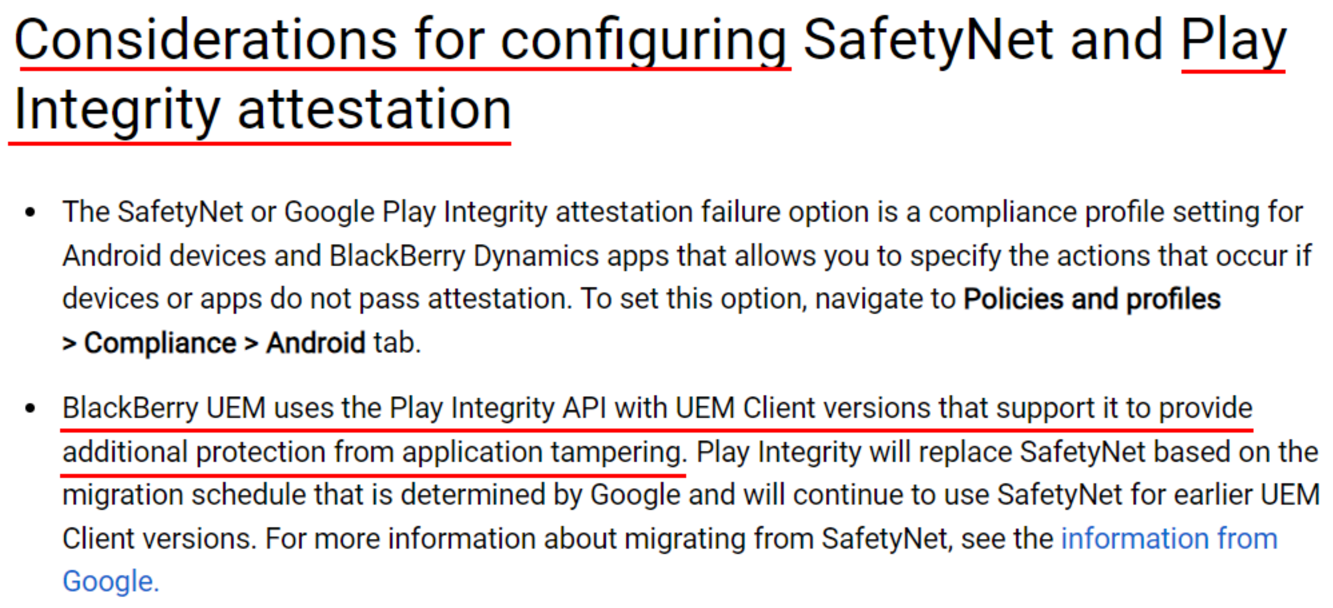


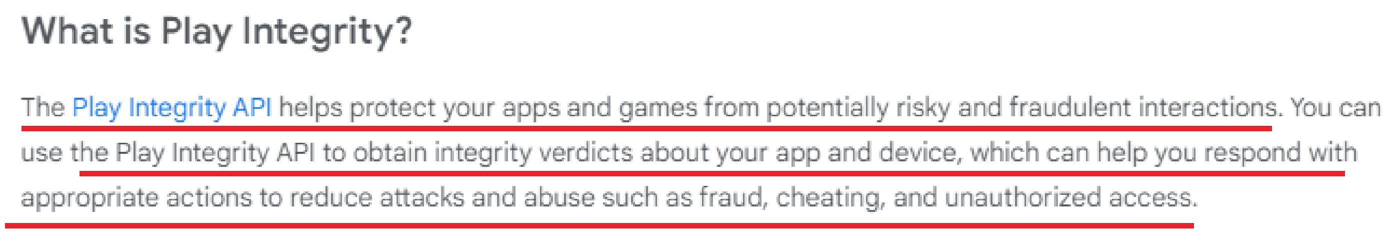
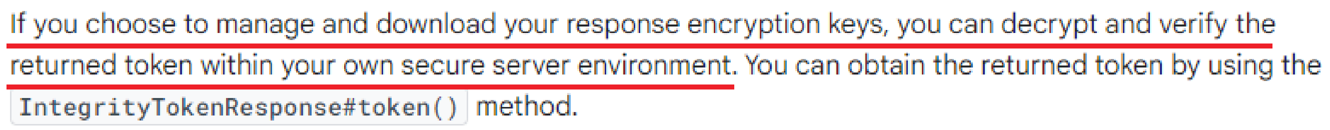
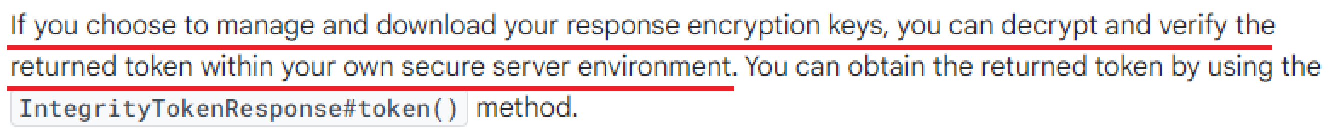
Configure attestation for Android devices and BlackBerry Dynamics apps

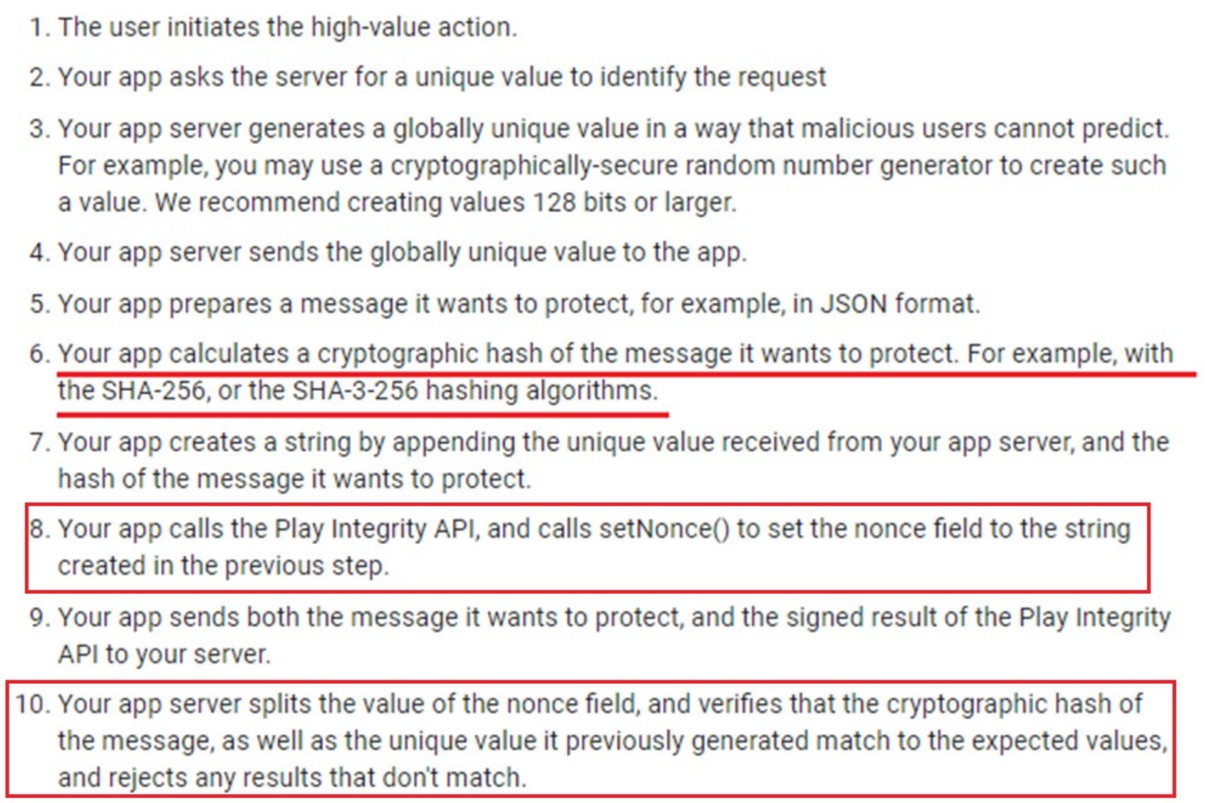
The latest version of Google Play services must be installed on users' devices.

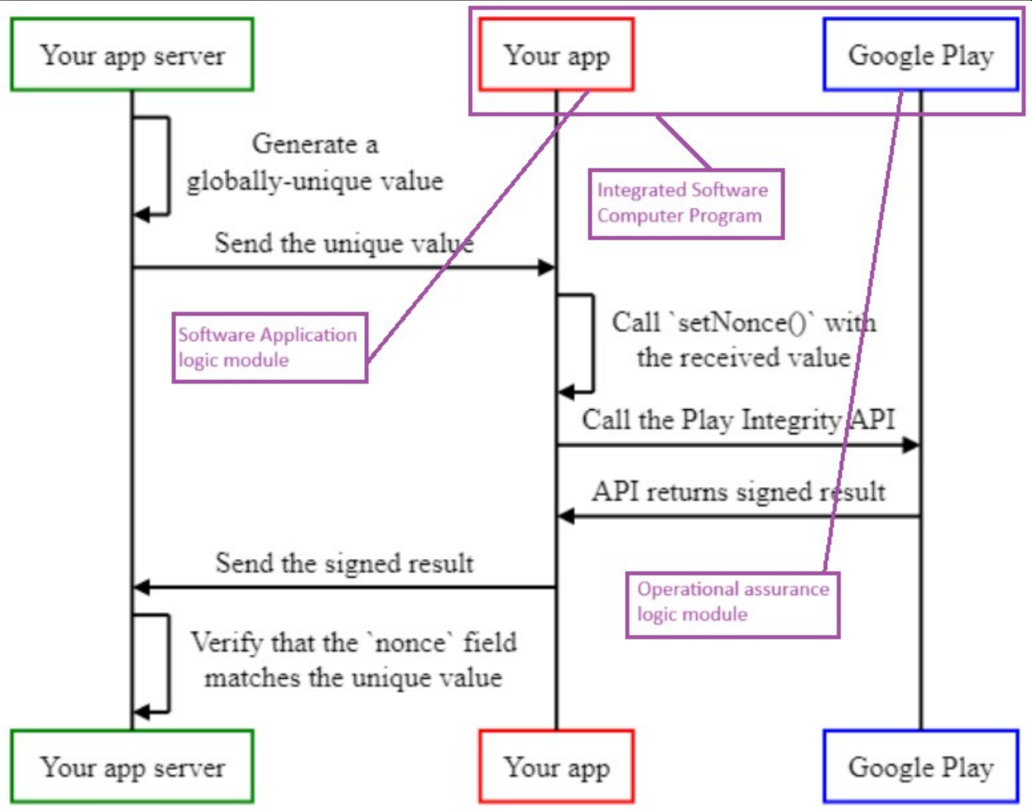
1. In the management console, on the menu bar, click **Settings > General settings > Attestation**.
2. Select the **Enable attestation challenges using SafetyNet or Play Integrity** check box.
3. If you want to enable the Google Compatibility Test Suite, select the **Enable CTS profile matching** check box.
4. In the **Challenge frequency** section, specify how often the device must return an attestation response to BlackBerry UEM. The default and minimum value is 24 hours.
5. In the **Grace period** section, specify the grace period for devices. When the grace period expires with no successful attestation response, a device is considered out of compliance and is subject to the actions that you specify in the assigned compliance profile.

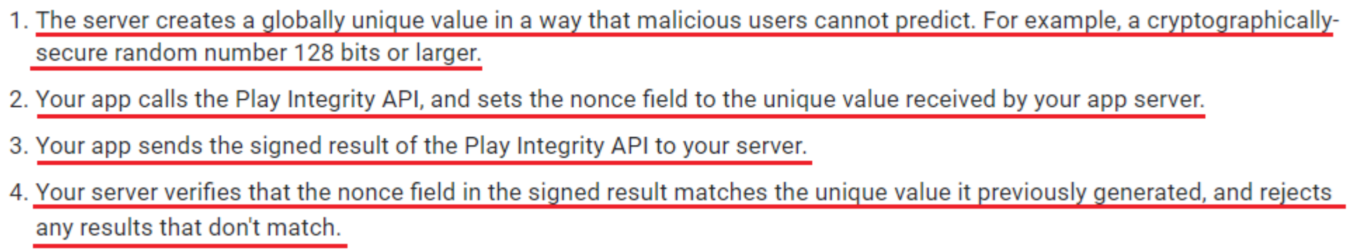
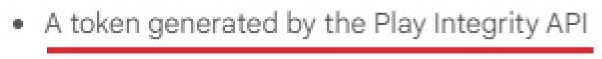
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841

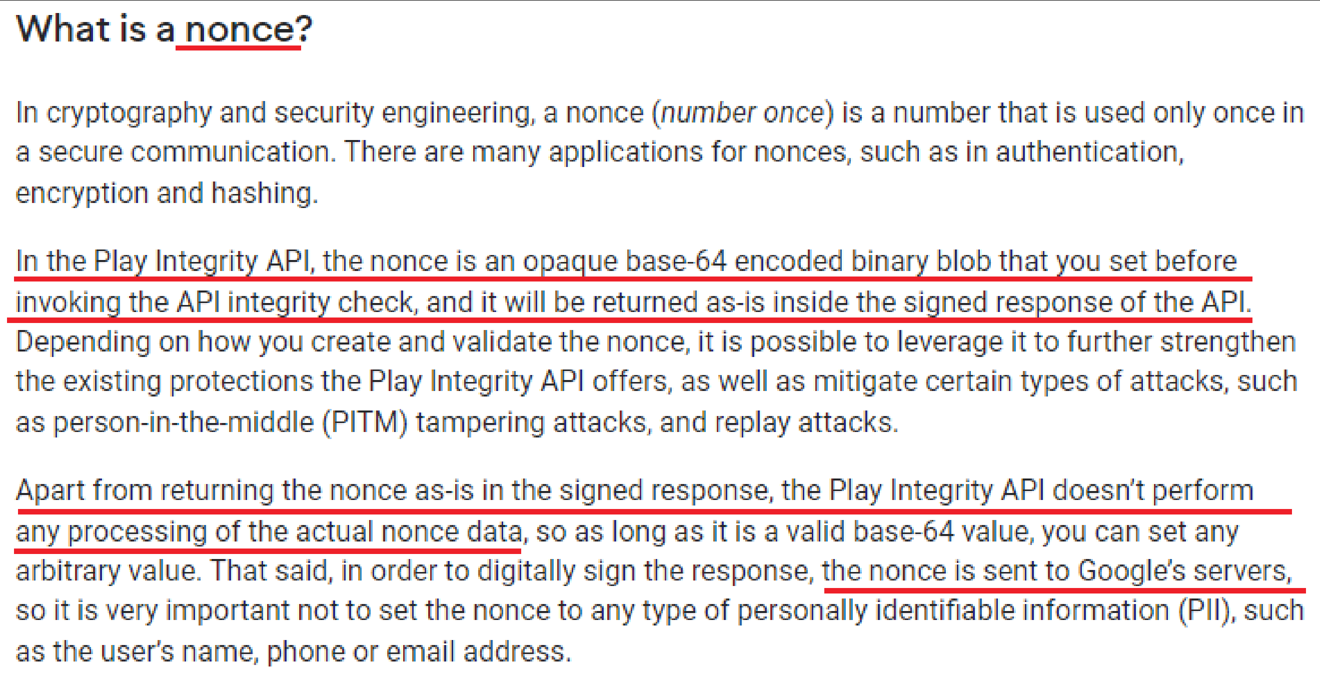
	 AFTER COMPLETING THE TASK • In the compliance profile assigned to devices, enable the "SafetyNet or Play Integrity attestation failure" rule and configure the actions that you want UEM to carry out when devices or BlackBerry Dynamics apps fail attestation. • In the management console, you can view a device's attestation status in the device details. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841  **Considerations for configuring SafetyNet and Play Integrity attestation** • The SafetyNet or Google Play Integrity attestation failure option is a compliance profile setting for Android devices and BlackBerry Dynamics apps that allows you to specify the actions that occur if devices or apps do not pass attestation. To set this option, navigate to **Policies and profiles > Compliance > Android** tab. • BlackBerry UEM uses the Play Integrity API with UEM Client versions that support it to provide additional protection from application tampering. Play Integrity will replace SafetyNet based on the migration schedule that is determined by Google and will continue to use SafetyNet for earlier UEM Client versions. For more information about migrating from SafetyNet, see the information from Google.

	https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_18/administration/device-features-it-policies/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/Considerations-for-configuring-attestation-for-Android-devices-and-apps-using-SafetyNet  **What is Play Integrity?** The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions. You can use the Play Integrity API to obtain integrity verdicts about your app and device, which can help you respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access. https://developer.android.com/codelabs/add-play-integrity#0  If you choose to manage and download your response encryption keys, you can decrypt and verify the returned token within your own secure server environment. You can obtain the returned token by using the `IntegrityTokenResponse#token()` method. https://developer.android.com/google/play/integrity/classic#nonce  If you choose to manage and download your response encryption keys, you can decrypt and verify the returned token within your own secure server environment. You can obtain the returned token by using the `IntegrityTokenResponse#token()` method. https://developer.android.com/google/play/integrity/classic#nonce

	 1. The user initiates the high-value action. 2. Your app asks the server for a unique value to identify the request 3. Your app server generates a globally unique value in a way that malicious users cannot predict. For example, you may use a cryptographically-secure random number generator to create such a value. We recommend creating values 128 bits or larger. 4. Your app server sends the globally unique value to the app. 5. Your app prepares a message it wants to protect, for example, in JSON format. 6. Your app calculates a cryptographic hash of the message it wants to protect. For example, with the SHA-256, or the SHA-3-256 hashing algorithms. 7. Your app creates a string by appending the unique value received from your app server, and the hash of the message it wants to protect. 8. Your app calls the Play Integrity API, and calls setNonce() to set the nonce field to the string created in the previous step. 9. Your app sends both the message it wants to protect, and the signed result of the Play Integrity API to your server. 10. Your app server splits the value of the nonce field, and verifies that the cryptographic hash of the message, as well as the unique value it previously generated match to the expected values, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

	 Your app server Your app Google Play Generate a globally-unique value Integrated Software Computer Program Send the unique value Software Application logic module Call `setNonce()` with the received value Call the Play Integrity API API returns signed result Send the signed result Operational assurance logic module Verify that the `nonce` field matches the unique value Your app server Your app Google Play https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1

	 1. The server creates a globally unique value in a way that malicious users cannot predict. For example, a cryptographically-secure random number 128 bits or larger. 2. Your app calls the Play Integrity API, and sets the nonce field to the unique value received by your app server. 3. Your app sends the signed result of the Play Integrity API to your server. 4. Your server verifies that the nonce field in the signed result matches the unique value it previously generated, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1  • A token generated by the Play Integrity API https://developer.android.com/codelabs/add-play-integrity#2



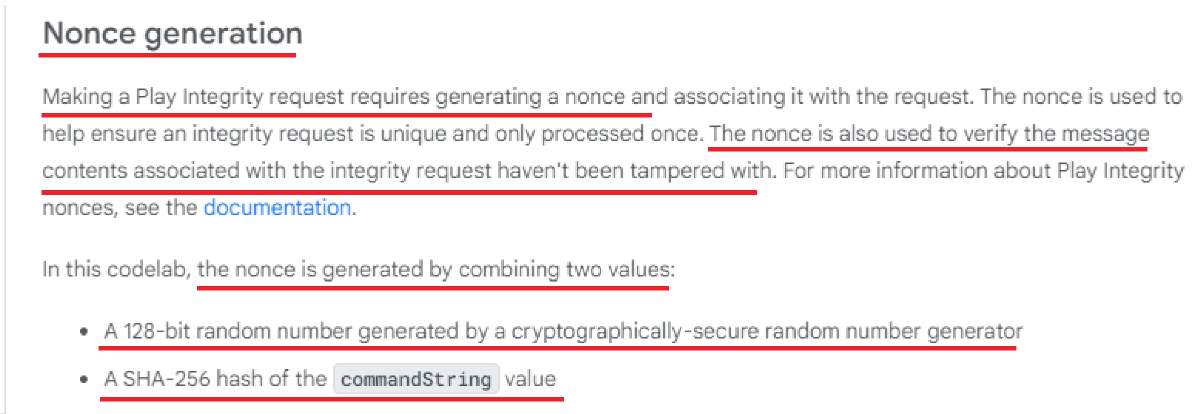
What is a nonce?

In cryptography and security engineering, a nonce (*number once*) is a number that is used only once in a secure communication. There are many applications for nonces, such as in authentication, encryption and hashing.

In the Play Integrity API, the nonce is an opaque base-64 encoded binary blob that you set before invoking the API integrity check, and it will be returned as-is inside the signed response of the API. Depending on how you create and validate the nonce, it is possible to leverage it to further strengthen the existing protections the Play Integrity API offers, as well as mitigate certain types of attacks, such as person-in-the-middle (PITM) tampering attacks, and replay attacks.

Apart from returning the nonce as-is in the signed response, the Play Integrity API doesn't perform any processing of the actual nonce data, so as long as it is a valid base-64 value, you can set any arbitrary value. That said, in order to digitally sign the response, the nonce is sent to Google's servers, so it is very important not to set the nonce to any type of personally identifiable information (PII), such as the user's name, phone or email address.

https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

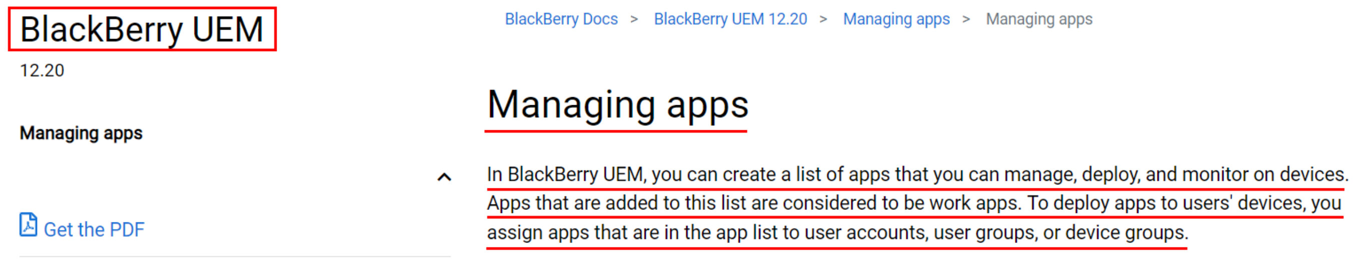


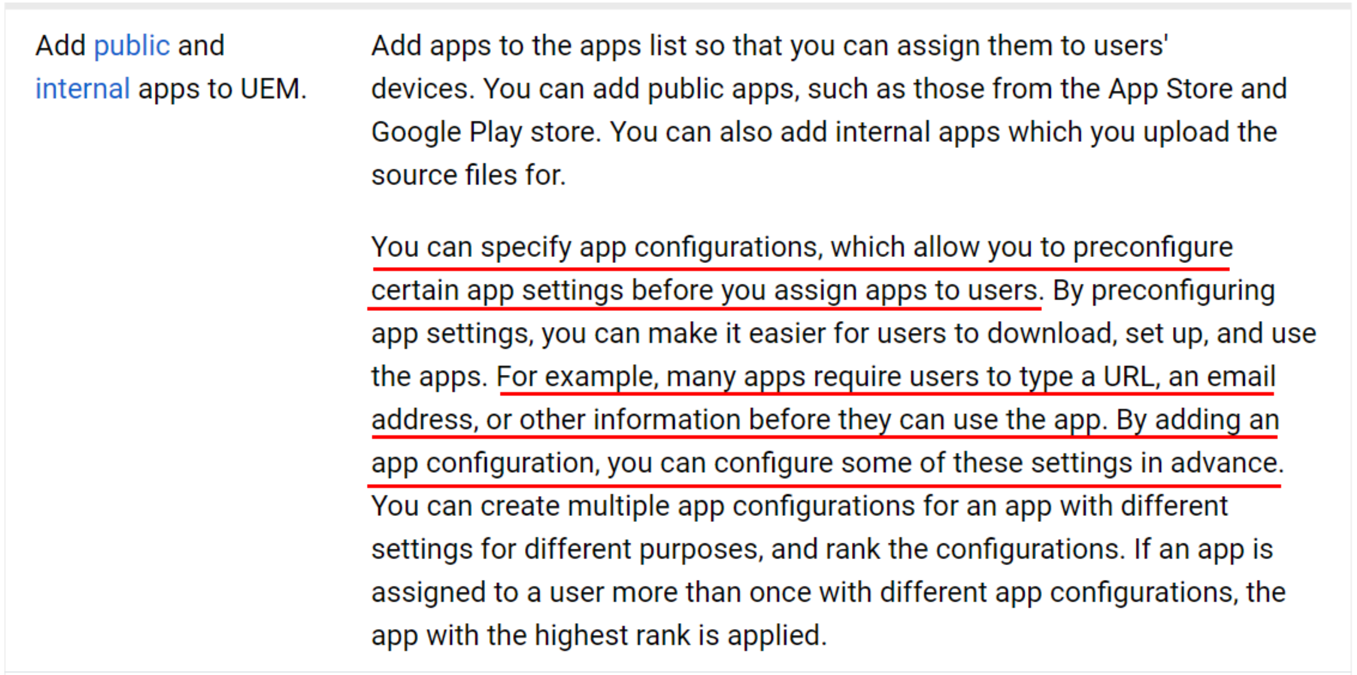
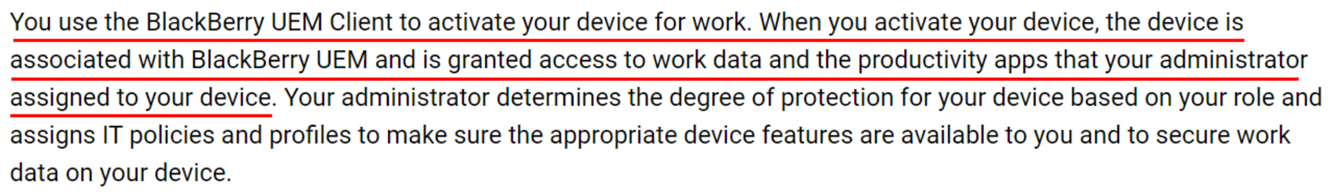
Nonce generation

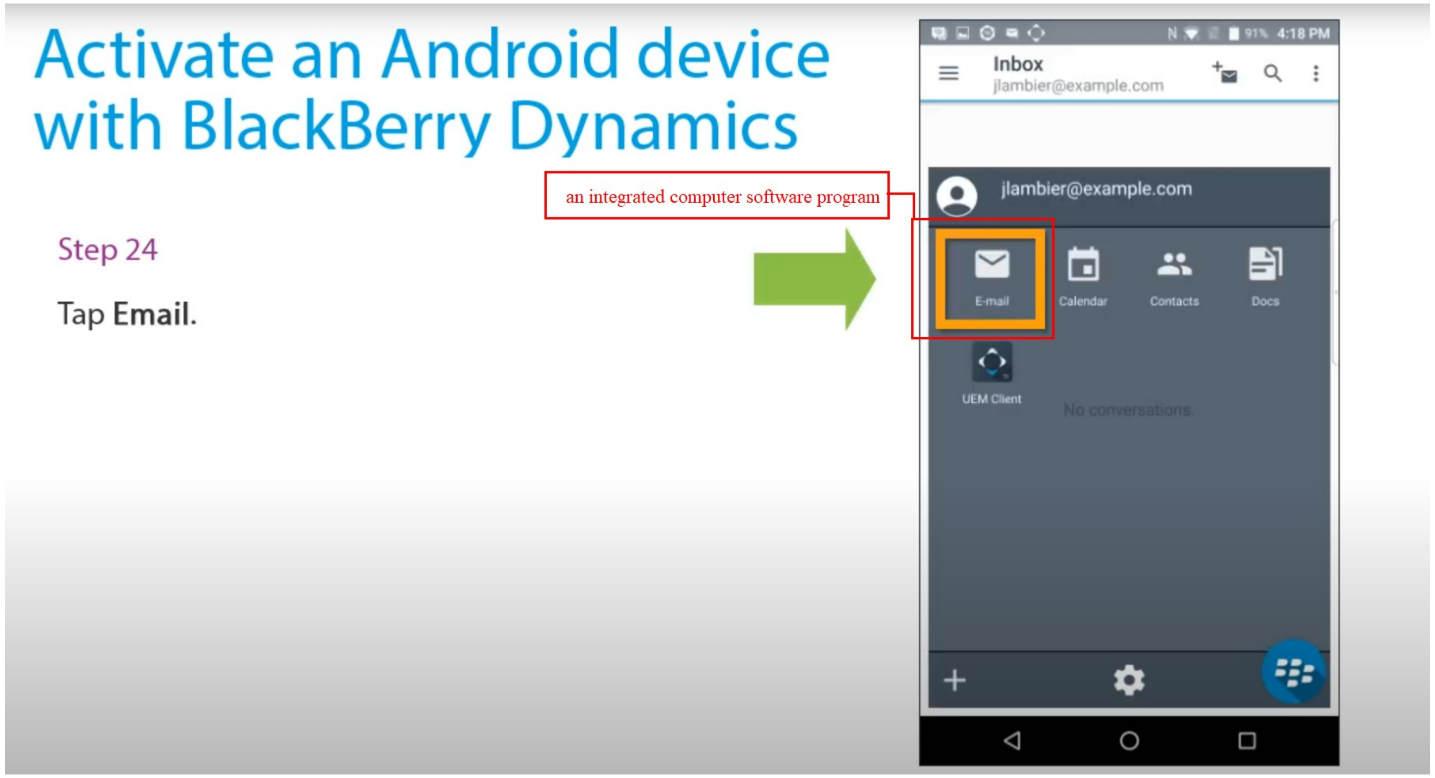
Making a Play Integrity request requires generating a nonce and associating it with the request. The nonce is used to help ensure an integrity request is unique and only processed once. The nonce is also used to verify the message contents associated with the integrity request haven't been tampered with. For more information about Play Integrity nonces, see the documentation.

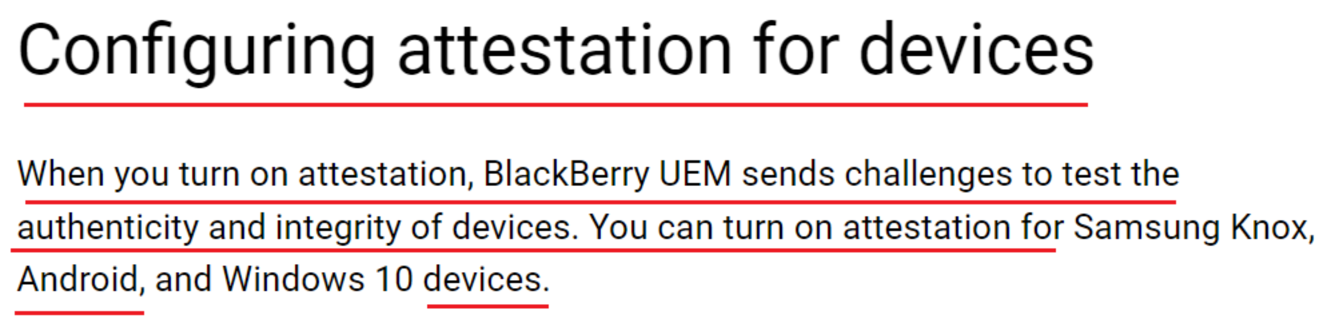
In this codelab, the nonce is generated by combining two values:

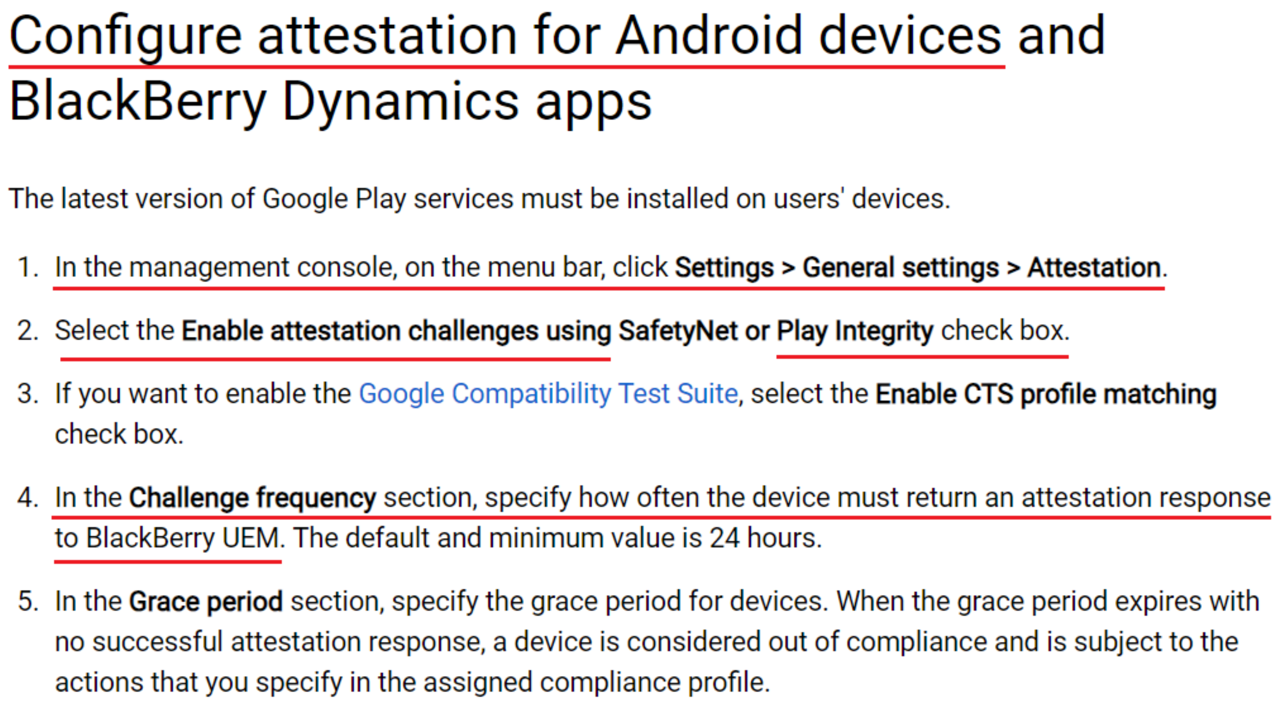
- A 128-bit random number generated by a cryptographically-secure random number generator
- A SHA-256 hash of the `commandString` value

	https://developer.android.com/codelabs/add-play-integrity#2
validating that the integrated software computer program as executed was not tampered with, responsive to the unique data tags.	The accused instrumentality discloses validating that the integrated software computer program (e.g., a work application along with attestation API, etc.) as executed was not tampered with, responsive to the unique data tags (e.g., a nonce, signed token, etc.). As shown below, the accused instrumentality is a Unified Endpoint Management (UEM) solution that allows enterprises to manage and control devices linked to employees. It allows administrators to add work applications that are permitted for use by employees on their managed devices. Also, the accused instrumentality utilizes device attestation for verifying device & app integrity. When execution of a work application (e.g., a software computer program) running on an employee device requires integrity verification, it utilizes device attestation feature for the verification. It utilizes an attestation API (e.g., an operation assurance logic module) such as Play Integrity API for an android device of an employee. The work app sends a request to the attestation API, it sends a signed token (e.g., unique security tag, etc.) back to the app. The app forwards the signed token to a server for verification. The server verifies whether the signed token matches its previously generated unique value. Thus, if the values match, it determines the valid app is executing a program on a valid device.  BlackBerry UEM 12.20 Managing apps Get the PDF BlackBerry Docs > BlackBerry UEM 12.20 > Managing apps > Managing apps Managing apps In BlackBerry UEM, you can create a list of apps that you can manage, deploy, and monitor on devices. Apps that are added to this list are considered to be work apps. To deploy apps to users' devices, you assign apps that are in the app list to user accounts, user groups, or device groups. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps

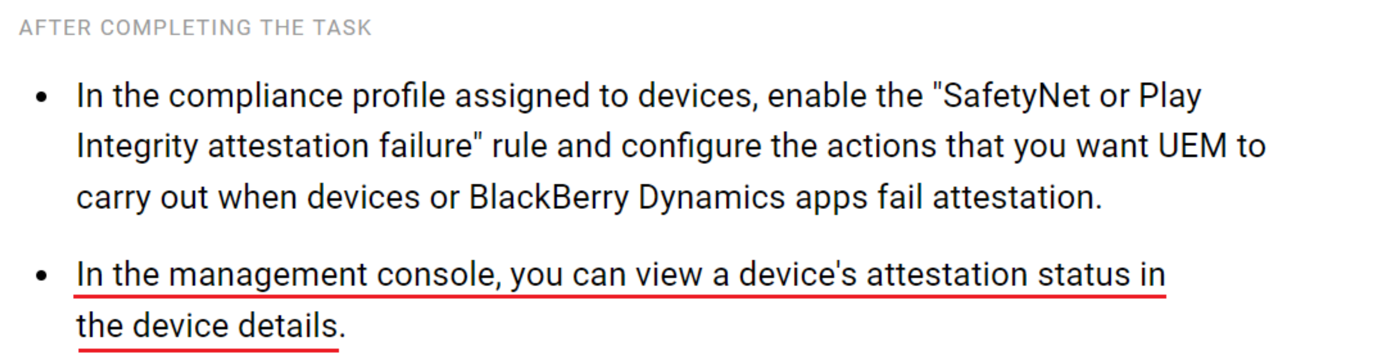
	 Add public and internal apps to UEM. Add apps to the apps list so that you can assign them to users' devices. You can add public apps, such as those from the App Store and Google Play store. You can also add internal apps which you upload the source files for. You can specify app configurations, which allow you to preconfigure certain app settings before you assign apps to users. By preconfiguring app settings, you can make it easier for users to download, set up, and use the apps. For example, many apps require users to type a URL, an email address, or other information before they can use the app. By adding an app configuration, you can configure some of these settings in advance. You can create multiple app configurations for an app with different settings for different purposes, and rank the configurations. If an app is assigned to a user more than once with different app configurations, the app with the highest rank is applied. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_20/managing-apps/managing-apps  You use the BlackBerry UEM Client to activate your device for work. When you activate your device, the device is associated with BlackBerry UEM and is granted access to work data and the productivity apps that your administrator assigned to your device. Your administrator determines the degree of protection for your device based on your role and assigns IT policies and profiles to make sure the appropriate device features are available to you and to secure work data on your device. https://docs.blackberry.com/en/endpoint-management/blackberry-uem-client/latest#:~:text=You%20use%20the%20BlackBerry%20UEM,work%20data%20on%20your%20device.

	 Activate an Android device with BlackBerry Dynamics Step 24 Tap **Email**. an integrated computer software program Inbox jlambier@example.com jlambier@example.com E-mail Calendar Contacts Docs UEM Client No conversations https://www.youtube.com/watch?v=zRrI1TpLVXM

	 Configuring attestation for devices When you turn on attestation, BlackBerry UEM sends challenges to test the authenticity and integrity of devices. You can turn on attestation for Samsung Knox, Android, and Windows 10 devices. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation



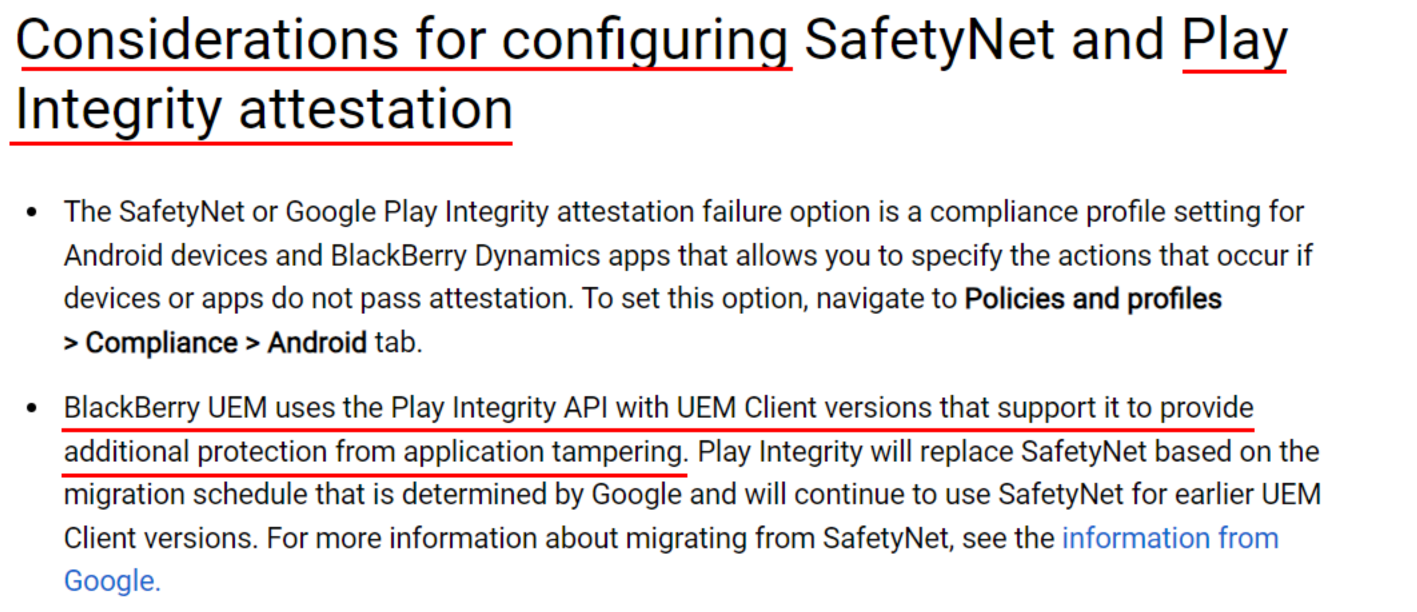
	Configure attestation for Android devices and BlackBerry Dynamics apps The latest version of Google Play services must be installed on users' devices. 1. In the management console, on the menu bar, click **Settings > General settings > Attestation**. 2. Select the **Enable attestation challenges using SafetyNet or Play Integrity** check box. 3. If you want to enable the Google Compatibility Test Suite, select the **Enable CTS profile matching** check box. 4. In the **Challenge frequency** section, specify how often the device must return an attestation response to BlackBerry UEM. The default and minimum value is 24 hours. 5. In the **Grace period** section, specify the grace period for devices. When the grace period expires with no successful attestation response, a device is considered out of compliance and is subject to the actions that you specify in the assigned compliance profile. https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841



AFTER COMPLETING THE TASK

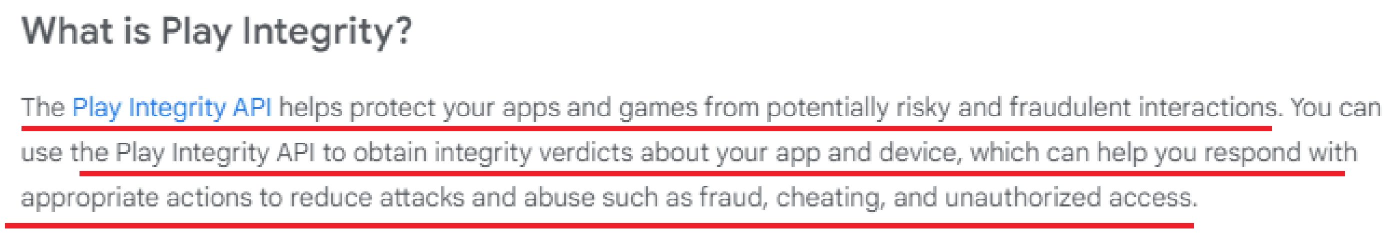
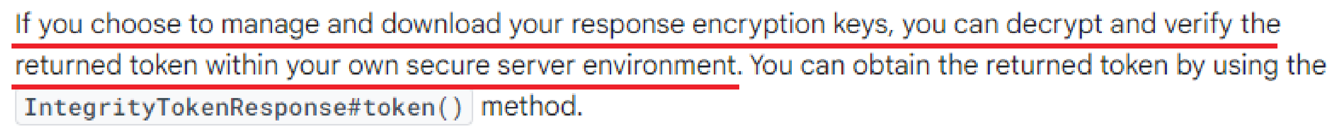
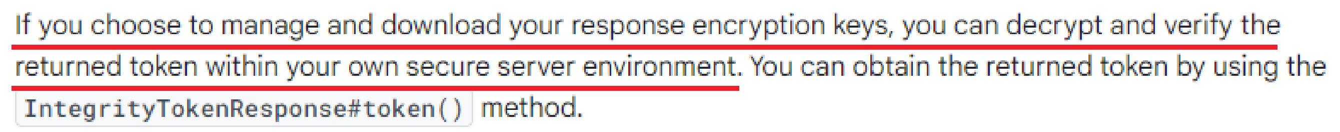
- In the compliance profile assigned to devices, enable the "SafetyNet or Play Integrity attestation failure" rule and configure the actions that you want UEM to carry out when devices or BlackBerry Dynamics apps fail attestation.
- In the management console, you can view a device's attestation status in the device details.

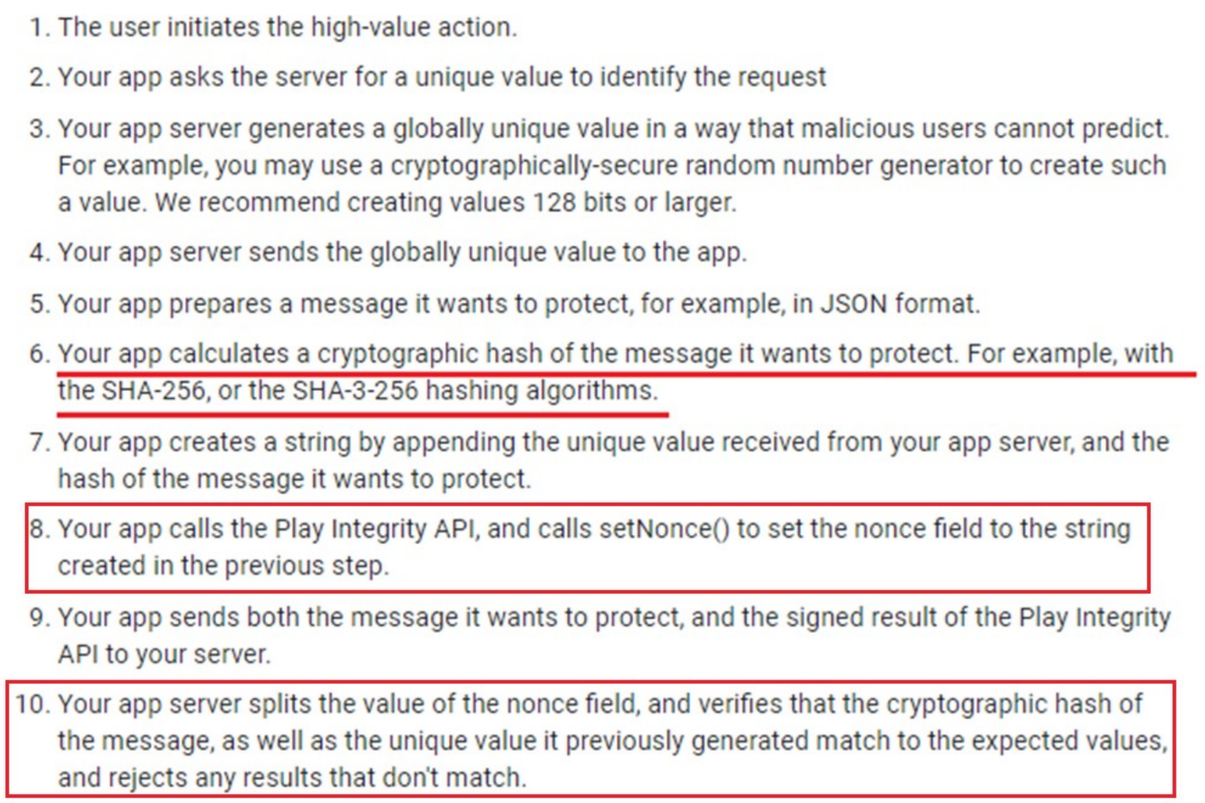
https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_19/device-configuration/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/hhd1523902719841

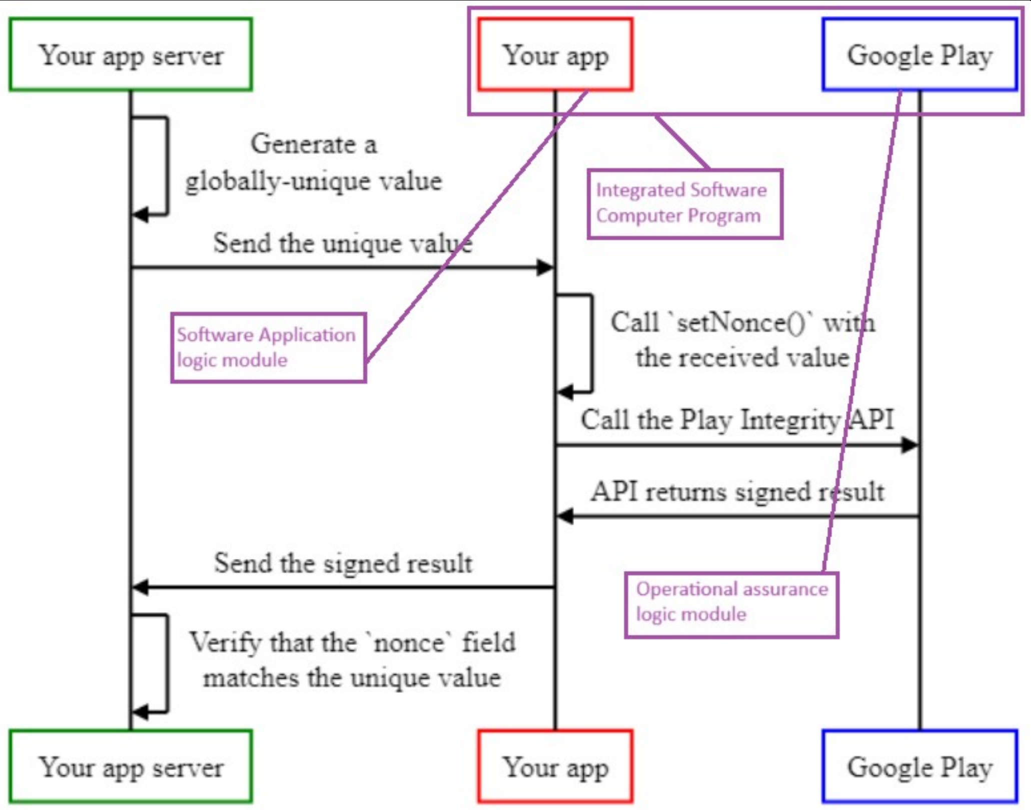


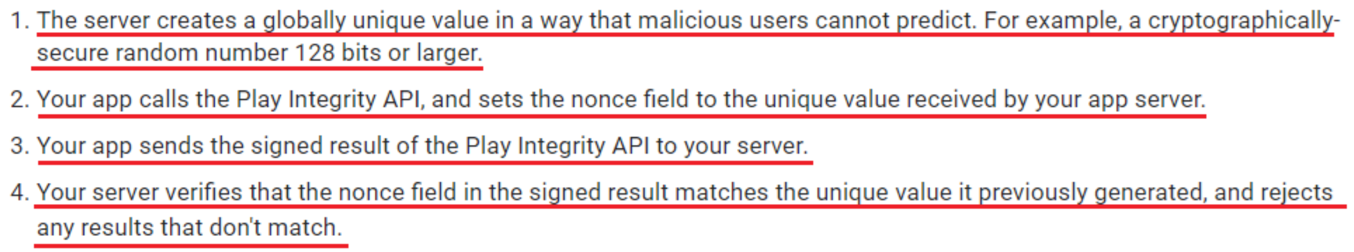
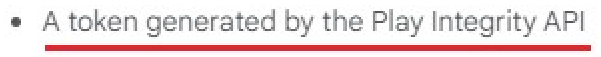
Considerations for configuring SafetyNet and Play Integrity attestation

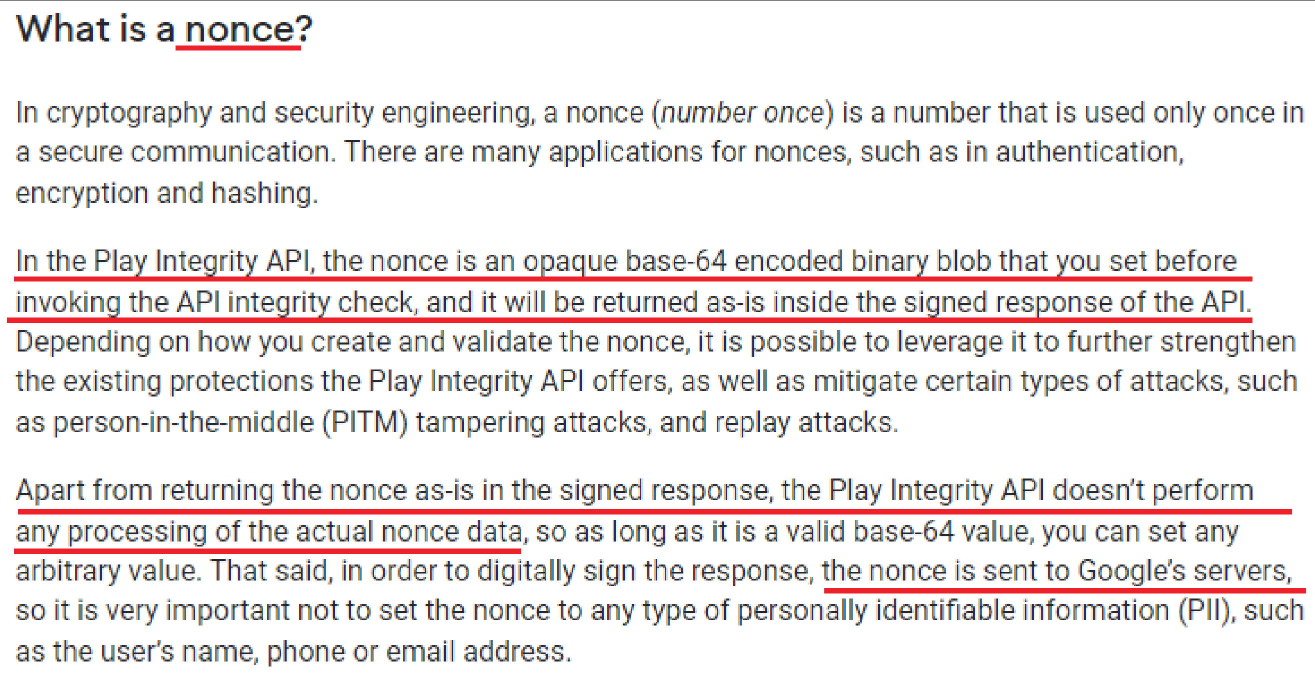
- The SafetyNet or Google Play Integrity attestation failure option is a compliance profile setting for Android devices and BlackBerry Dynamics apps that allows you to specify the actions that occur if devices or apps do not pass attestation. To set this option, navigate to **Policies and profiles > Compliance > Android** tab.
- BlackBerry UEM uses the Play Integrity API with UEM Client versions that support it to provide additional protection from application tampering. Play Integrity will replace SafetyNet based on the migration schedule that is determined by Google and will continue to use SafetyNet for earlier UEM Client versions. For more information about migrating from SafetyNet, see the information from Google.

	https://docs.blackberry.com/en/endpoint-management/blackberry-uem/12_18/administration/device-features-it-policies/managing-attestation/Configure-attestation-for-Android-devices-and-apps-using-SafetyNet/Considerations-for-configuring-attestation-for-Android-devices-and-apps-using-SafetyNet  **What is Play Integrity?** The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions. You can use the Play Integrity API to obtain integrity verdicts about your app and device, which can help you respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access. https://developer.android.com/codelabs/add-play-integrity#0  If you choose to manage and download your response encryption keys, you can decrypt and verify the returned token within your own secure server environment. You can obtain the returned token by using the `IntegrityTokenResponse#token()` method. https://developer.android.com/google/play/integrity/classic#nonce  If you choose to manage and download your response encryption keys, you can decrypt and verify the returned token within your own secure server environment. You can obtain the returned token by using the `IntegrityTokenResponse#token()` method. https://developer.android.com/google/play/integrity/classic#nonce

	 1. The user initiates the high-value action. 2. Your app asks the server for a unique value to identify the request 3. Your app server generates a globally unique value in a way that malicious users cannot predict. For example, you may use a cryptographically-secure random number generator to create such a value. We recommend creating values 128 bits or larger. 4. Your app server sends the globally unique value to the app. 5. Your app prepares a message it wants to protect, for example, in JSON format. 6. Your app calculates a cryptographic hash of the message it wants to protect. For example, with the SHA-256, or the SHA-3-256 hashing algorithms. 7. Your app creates a string by appending the unique value received from your app server, and the hash of the message it wants to protect. 8. Your app calls the Play Integrity API, and calls setNonce() to set the nonce field to the string created in the previous step. 9. Your app sends both the message it wants to protect, and the signed result of the Play Integrity API to your server. 10. Your app server splits the value of the nonce field, and verifies that the cryptographic hash of the message, as well as the unique value it previously generated match to the expected values, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html

	 Your app server Your app Google Play Generate a globally-unique value Integrated Software Computer Program Send the unique value Software Application logic module Call `setNonce()` with the received value Call the Play Integrity API API returns signed result Send the signed result Operational assurance logic module Verify that the `nonce` field matches the unique value Your app server Your app Google Play https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1

	 1. The server creates a globally unique value in a way that malicious users cannot predict. For example, a cryptographically-secure random number 128 bits or larger. 2. Your app calls the Play Integrity API, and sets the nonce field to the unique value received by your app server. 3. Your app sends the signed result of the Play Integrity API to your server. 4. Your server verifies that the nonce field in the signed result matches the unique value it previously generated, and rejects any results that don't match. https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html?m=1  • A token generated by the Play Integrity API https://developer.android.com/codelabs/add-play-integrity#2



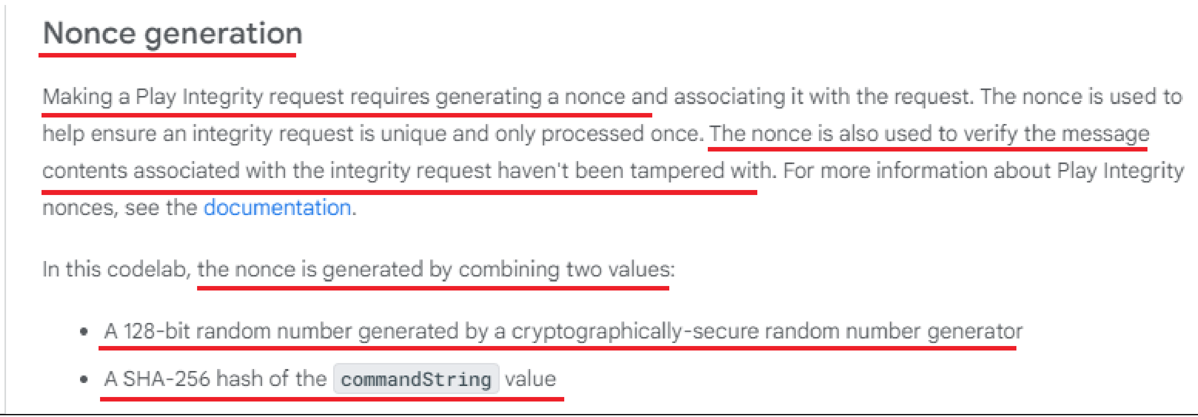
What is a nonce?

In cryptography and security engineering, a nonce (*number once*) is a number that is used only once in a secure communication. There are many applications for nonces, such as in authentication, encryption and hashing.

In the Play Integrity API, the nonce is an opaque base-64 encoded binary blob that you set before invoking the API integrity check, and it will be returned as-is inside the signed response of the API. Depending on how you create and validate the nonce, it is possible to leverage it to further strengthen the existing protections the Play Integrity API offers, as well as mitigate certain types of attacks, such as person-in-the-middle (PITM) tampering attacks, and replay attacks.

Apart from returning the nonce as-is in the signed response, the Play Integrity API doesn't perform any processing of the actual nonce data, so as long as it is a valid base-64 value, you can set any arbitrary value. That said, in order to digitally sign the response, the nonce is sent to Google's servers, so it is very important not to set the nonce to any type of personally identifiable information (PII), such as the user's name, phone or email address.

https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html



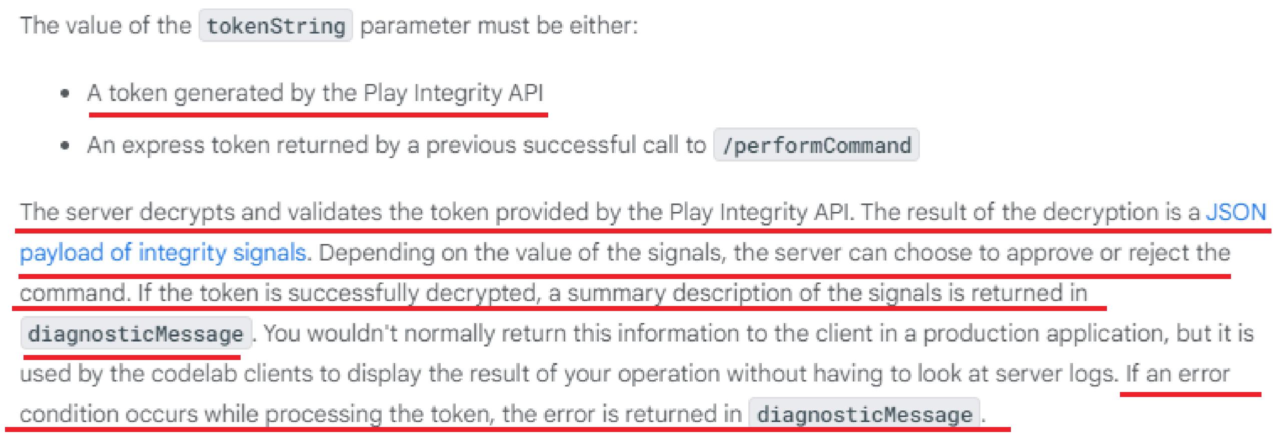
Nonce generation

Making a Play Integrity request requires generating a nonce and associating it with the request. The nonce is used to help ensure an integrity request is unique and only processed once. The nonce is also used to verify the message contents associated with the integrity request haven't been tampered with. For more information about Play Integrity nonces, see the documentation.

In this codelab, the nonce is generated by combining two values:

- A 128-bit random number generated by a cryptographically-secure random number generator
- A SHA-256 hash of the `commandString` value

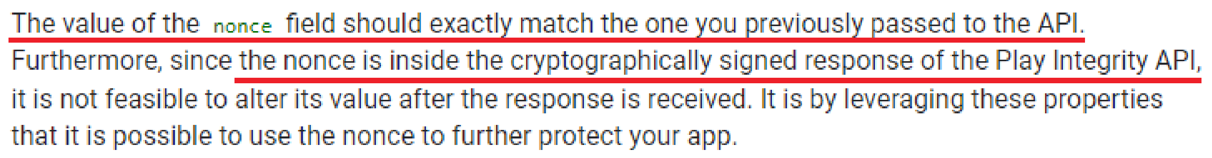
https://developer.android.com/codelabs/add-play-integrity#2


The value of the `tokenString` parameter must be either:

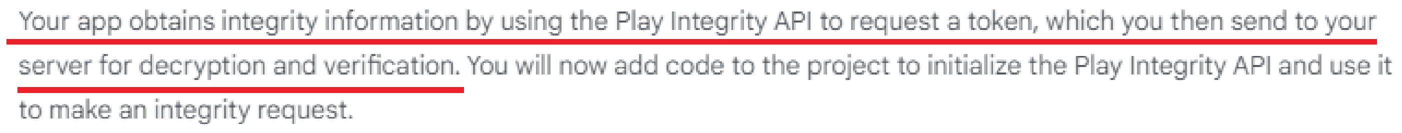
- A token generated by the Play Integrity API
- An express token returned by a previous successful call to `/performCommand`

The server decrypts and validates the token provided by the Play Integrity API. The result of the decryption is a JSON payload of integrity signals. Depending on the value of the signals, the server can choose to approve or reject the command. If the token is successfully decrypted, a summary description of the signals is returned in `diagnosticMessage`. You wouldn't normally return this information to the client in a production application, but it is used by the codelab clients to display the result of your operation without having to look at server logs. If an error condition occurs while processing the token, the error is returned in `diagnosticMessage`.

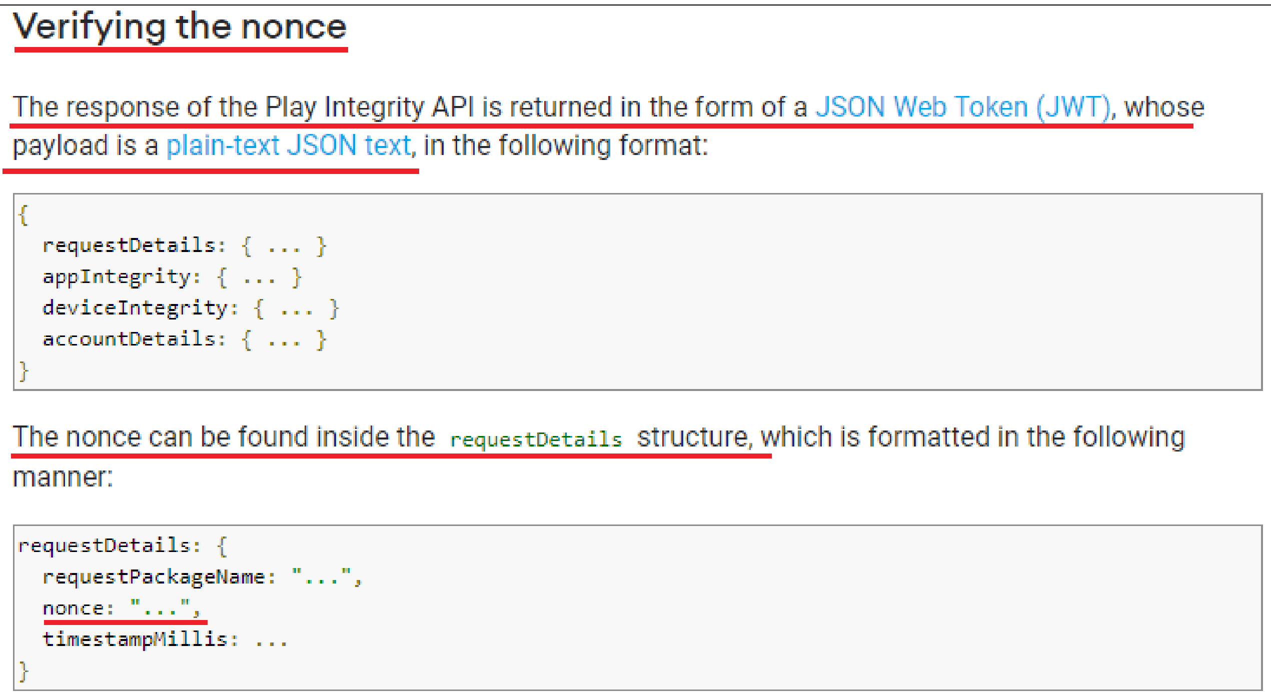
https://developer.android.com/codelabs/add-play-integrity#2


The value of the `nonce` field should exactly match the one you previously passed to the API. Furthermore, since the nonce is inside the cryptographically signed response of the Play Integrity API, it is not feasible to alter its value after the response is received. It is by leveraging these properties that it is possible to use the nonce to further protect your app.

https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html


Your app obtains integrity information by using the Play Integrity API to request a token, which you then send to your server for decryption and verification. You will now add code to the project to initialize the Play Integrity API and use it to make an integrity request.

https://developer.android.com/codelabs/add-play-integrity#8



Verifying the nonce

The response of the Play Integrity API is returned in the form of a JSON Web Token (JWT), whose payload is a plain-text JSON text, in the following format:

```
{
  requestDetails: { ... }
  appIntegrity: { ... }
  deviceIntegrity: { ... }
  accountDetails: { ... }
}
```

The nonce can be found inside the `requestDetails` structure, which is formatted in the following manner:

```
requestDetails: {
  requestPackageName: "...",
  nonce: "...",
  timestampMillis: ...
}
```

https://android-developers.googleblog.com/2022/05/boost-security-of-your-app-with-nonce.html